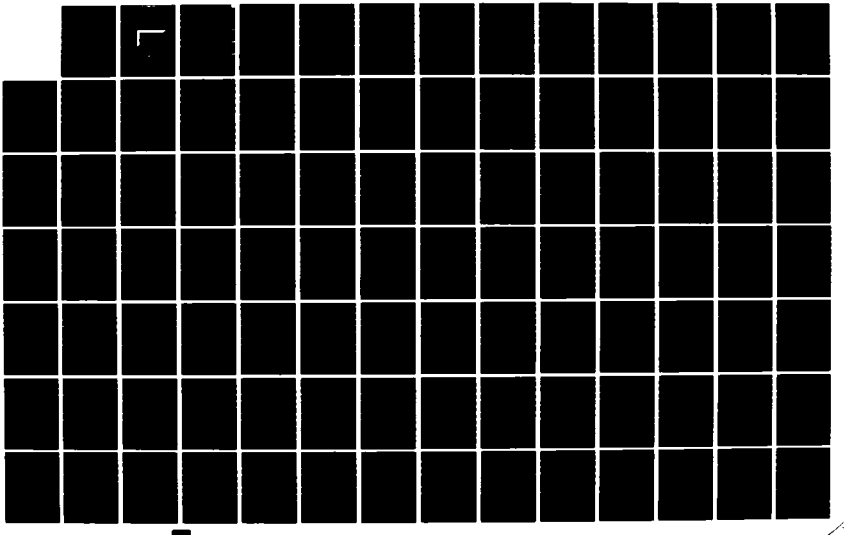
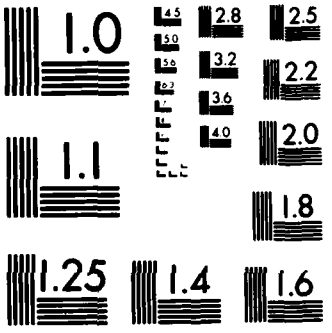
AD-A150 840 A TWO-DIMENSIONAL DESIGN METHOD FOR HIGHLY-LOADED BLADES IN TURBOMACHINES.. (U) MASSACHUSETTS INST OF TECH CAMBRIDGE GAS TURBINE AND PLASMA D.. T Q DANG ET AL. 1/2

UNCLASSIFIED APR 83 GT/PDL-173 AFOSR-TR-85-0066 F/G 20/4 NL





A TWO-DIMENSIONAL DESIGN METHOD FOR HIGHLY-LOADED BLADES IN TURBOMACHINES

by

Thong Quoc Dang

James E. McCune

GT & PDL Report No. 173 April 1983

GAS TURBINE & PLASMA DYNAMICS LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS

A TWO-DIMENSIONAL DESIGN METHOD FOR HIGHLY-LOADED BLADES IN TURBOMACHINES

by

Thong Quoc Dang

James E. McCune

GT & PDL Report No. 173

April 1983

This research was supported by the U.S. Air Force Office of Scientific Research, Contract Number F49620-82-K-0002, Dr. J. D. Wilson, Program Manager.

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution Unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S) [illegible] No. 173	5 MONITORING ORGANIZATION REPORT NUMBER(S) AFOSR-TR- 85-0066
6a NAME OF PERFORMING ORGANIZATION [illegible]	6b OFFICE SYMBOL (If applicable)
7a NAME OF MONITORING ORGANIZATION AFOSR/NA
6c ADDRESS (City, State and ZIP Code) [illegible] LABORATORY [illegible]	7b ADDRESS (City, State and ZIP Code) [illegible]
8a NAME OF FUNDING/SPONSORING ORGANIZATION AIR FORCE OFFICE OF SCIENTIFIC RESEARCH	8b OFFICE SYMBOL (If applicable) AFOSR/NA
	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F49620-82-K-0002
8c ADDRESS (City, State and ZIP Code) [illegible], DC [illegible]	10 SOURCE OF FUNDING NOS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT NO
61102F	2307	A4	

11 TITLE (Include Security Classification) TWO-DIMENSIONAL [illegible] BLADES IN TURBOMACHINES (UNCLASSIFIED)

12 PERSONAL AUTHOR(S)
C. MCCUNE, JAMES E

13a TYPE OF REPORT	13b TIME COVERED FROM ___ TO ___	14 DATE OF REPORT (Yr, Mo, Day)	15 PAGE COUNT
		1983, April	145

16 SUPPLEMENTARY NOTATION

17 COSATI CODES FIELD	GROUP	SUB GR

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
TURBOMACHINERY COMPUTATIONAL FLUID DYNAMICS
CASCADES
INVISID FLOW

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

A [illegible] design method for high-loaded blades in an isolated cascade is presented [illegible] blades. The flow is assumed to be incompressible and inviscid. The upstream [illegible] flow condition is taken to be uniform. The present goals of this research are [illegible] a practical numerical code for the design problem, and a nonlinear theory [illegible] can be easily expanded to three-dimensions. The theory is based in part on the [illegible] formulation. The blade profile is determined iteratively through the blade [illegible] boundary conditions using a "smoothing" technique". The program gives very fast [illegible] solutions with satisfactory accuracy for practical solidity range.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
[illegible] UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS ☐	UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE NUMBER (Include Area Code)	22c OFFICE SYMBOL
	202/767-4935	AFOSR/NA

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

ABSTRACT

A practical design method for highly-loaded blades in an isolated cascade is presented in this thesis. The flow is assumed to be incompressible and inviscid. The upstream inlet flow condition is taken to be uniform. The present goals of this research are to provide a practical numerical code for the design problem, and a non-linear theory which can be easily expanded to three-dimensions. The theory is based in part on the Clebsh formulation. The blade profile is determined iteratively through the blade boundary conditions using a "smoothing" technique. A practical numerical code is presented for the design problem using "partial smoothing". The program gives very fast convergence solutions with satisfactory accuracy for practical solidity range.

Acknowledgements

I wish to extend my most sincere thanks to Professor J.E. McCune, my thesis supervisor, for his patient supervision at the beginning of this work, and his continuing guidance, support and encouragement.

I also wish to express my thanks to the following people for their support in preparing this thesis:

- Dr. C.S. Tan for his guidance in the numerical work and many other matters.
- Mr. E.R. McFarland of NASA Lewis Research Center for providing his direct method numerical code in great details.
- Professor W.R. Hawthorne for discussing the project.

Many thanks to my family for their supports during my staying at MIT.

This research was supported by the U.S. Air Force Office of Scientific Research, Contract Number F49620-82-K-0002, Dr. J.D. Wilson, Program Manager.

Figure 1 was copied from reference [4].

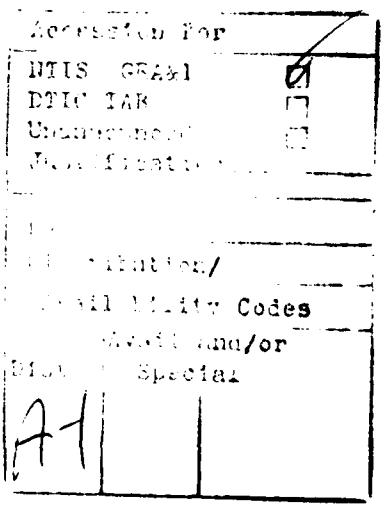


Table of Contents

List of Tables

List of Figures

Partial List of Symbols

Δ_p - pressure coefficient

f - blade camber line ordinate

f_o - mean streamline ordinate

G - negative of the y-component of the pitch average velocity

G_T - negative of the y-component of the gap average velocity

I - "smoothing" function (defined in Appendix C)

J - "smoothing" function (defined in Appendix C)

K - "smoothing" function (defined in Appendix C)

λ - spacing to chord ratio

S - "sawtooth" function (defined in Appendix C)

T - blockage distribution

V - actual velocity

$\bar{V}$ - average velocity

$\tilde{v}$ - perturbed velocity

α - blade camber surfaces

α_1 - inlet angle

α_2 - outlet angle

δ_p - "periodic delta" function (defined in Appendix C)

Ω - actual vorticity

$\bar{\Omega}$ - average vorticity

$\tilde{\Omega}$ - perturbed vorticity

$(\ )^+$ - pressure side

$(\ )^-$ - suction side

$(\ )_x$ - x-compenent of a vector

$(\ )_y$ - y-component of a vector

NOTATIONS

$$\overline{(\ )} \equiv \frac{1}{s}\int_f^{f+s} (\ )\, dy$$

$$\overline{(\ )}_T \equiv \frac{1}{s-2T}\int_{f+T}^{f+s-T} (\ )\, dy$$

$$<(\ )> \equiv \frac{1}{2}\left[(\ )^{f^+} + (\ )^{f^-}\right]$$

$$<(\ )>_T \equiv \frac{1}{2}\left[(\ )^{f^+_{+T}} + (\ )^{f^-_{-T}}\right]$$

$$\Delta_T(\ ) \equiv \frac{1}{2}\left[(\ )^{f^+_{+T}} - (\ )^{f^-_{-T}}\right]$$

Chapter 1 : Introduction

This thesis presents a technique for solving the design problem (the inverse problem) for highly loaded blades in an isolated cascade. In the present study, the flow is treated as if it were incompressible and inviscid, and the upstream inlet flow condition is assumed to be uniform.

Historically, there have been several approaches to the design problem. In one approach, for example, the velocity distributions on both surfaces of the blade are specified, and the resulting blade shape is calculated. The advantage of this technique is that the designer can prescribe surface pressure distributions which minimize the chance of flow separation. However, the resulting blade geometry is not guaranteed to have a realistic configuration: the blade may be wavy, or even have negative thickness.

In a second approach, the velocity distribution on one surface and the profile thickness distribution are specified, and the resulting blade shape is calculated. This formulation does not guarantee obtaining an acceptable pressure distribution on the second blade surface. Moreover, the resulting loading distribution may not be structurally favorable: for example, the loading may be maximum where the blade is thinnest.

In a third approach, the loading and thickness (or 'blockage') distributions are specified, and the resulting blade shape is calculated. This formulation does not guarantee giving acceptable pressure distributions on either of the blade surfaces, but the resulting blade shape can be guaranteed to be at least structurally sound.

In applying classical aerodynamics methods to these problems, the presence of the blades and their effects on the flow can be modeled by distributing singularities (vortices, sources and sinks) on the blade camber or blade surfaces. Lewis [1] carried out the design problem using the first two approaches mentioned above. His analysis is based on Martensen's method [2]: vortices are distributed on each of the blade surfaces, and the induced flow field in the cascade plane is calculated using the Biot-Savart law.

Kashiwabara [3] carried out the design problem using the first of the above approaches by arranging vortices, sources and sinks along the blade camber. This theory also attempts to take into account some three-dimensional effects, and can be used for designing blades in axial, mixed and radial turbomachines.

These design theories, based in part on the Biot-Savart law, are simple for cascade calculation only. Until recently [4], only linearized theories have been developed to design three-dimensional blades [5], [6].

One goal of the present project is to provide a practical numerical code for the design problem which can give good accuracy and fast convergence solutions. An equally important goal is to provide a non-linear theory which can be practically expanded to three-dimensions. The present theory is based in part on the Clebsh formulation which has been successfully investigated by Tan, Wang, Hawthorne and McCune [4] in their study of a three-dimensionel design method for highly-loaded but infinitely thin blades.

In the present study, the blade profile is determined iteratively through the blade boundary conditions using a "smoothing" technique: the velocity potential is expressed in a series of "smoothing" functions developed by

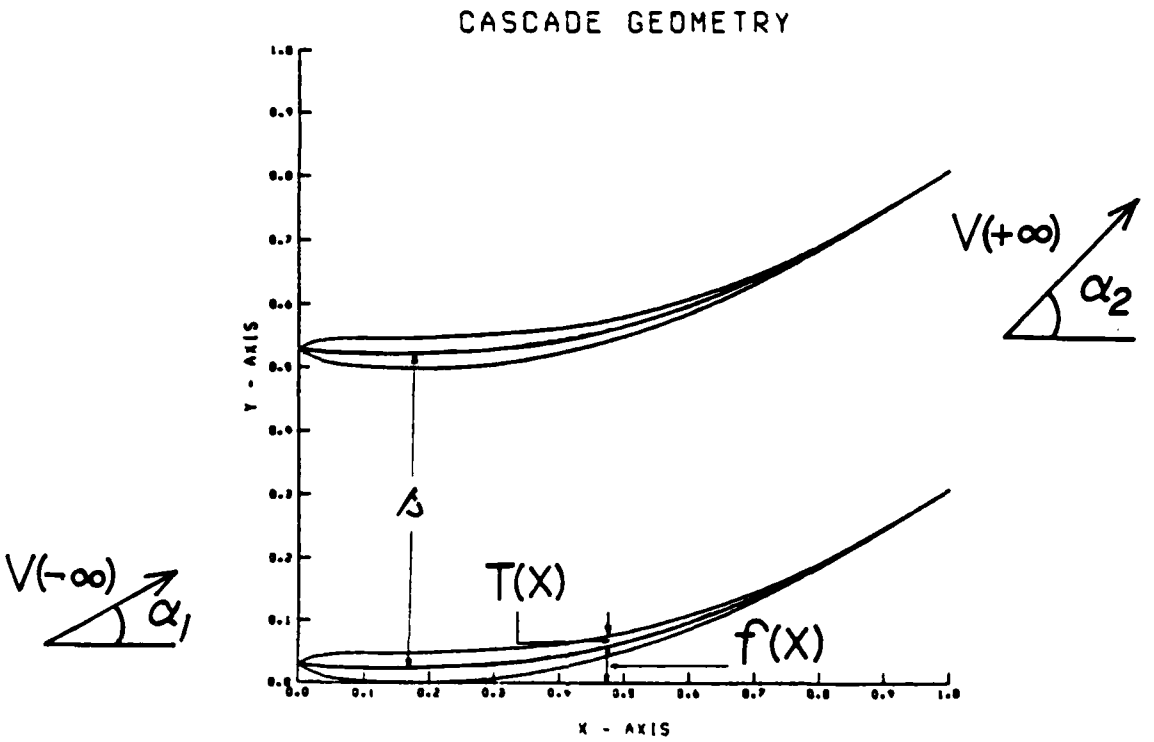
McCune [Appendix C]. The "smoothing" technique represents an asymptotic expansion (in the blade spacing) of the Green's function for blade rows, and can be applied in both 2D and 3D.

In the first example, the blades are assumed to be infinitely thin (the 'Zero-Thickness' problem) to show the power of the "smoothing" technique: the blade shape is solved iteratively through a set of algebriac equations. The results compare very well with the "exact" method [4]. Then, a similar approach is used to solve the inverse problem (in the third formulation, outlined above) for high swirl blades having prescribed finite thickness (the 'Finite-Thickness' problem). A set of numerical examples are presented, which in part use "partial smoothing", i.e. a practical truncation of the smoothing series. Blades shapes with prescribed thickness (or blockage) distributions are obtained for prescribed swirl schedules or loading distributions. To confirm the results a 'direct' method , developed by McFarland [10], is used to compute both the circulation and the pressure coefficients on the blade obtained from the indirect method. The results show that the blade does in fact produce the desired circulation and loading distribution. Moreover, the pressure distributions on each surface agree well, at least away from the leading and trailing edges.

Finally, a practical design procedure is presented which allows for rapid exploration of various blade loading and thickness distributions, so as to be able to select favorable individual-surface pressure distributions as well.

Chapter 2 : Theoretical approach

2.1 Cascade geometry



The cascade geometry is shown in the above figure. The flow direction is from left to right. All lengths are non-dimensionalized to the axial chord length.

The blade camber lines are located at:

where n = 0,1,2,3, ...

s = spacing between blade cambers

f = location of the camber line n = 0

If we define α as

$$\alpha = y - f(x)$$

then blade cambers lie on surfaces of $\alpha = ns$ $\quad n = 0, \pm 1, \pm 2 \ldots$

Consider the blade located at $n = 0$. The blockage distribution $T(x)$ is defined such that:

- the blade upper surface is located at $y = f(x) + T(x)$
- the blade lower surface is located at $y = f(x) - T(x)$

Physically, $2T(x)$ can be interpreted as the axial blockage distribution. We note that $T(x)$ is not the blade thickness as defined in classical aerodynamics, where the thickness is defined as the perpendicular distance from the blade camber to the blade surface.

The far upstream flow condition is assumed to be uniform. All velocities are non-dimensionalized to the upstream inlet x-component velocity. The inlet flow angle is denoted by α_1. Likewise, the outlet flow angle is denoted by α_2.

2.2 Fluid mechanics background

As mentioned in the introduction, we have chosen to model the presence of the blade shape by distributing singularities on the blade camber. In this section, we will derive the governing equation for the velocity field in the "Zero-Thickness" problem. In chapter 4, we will extend this theory to the "Finite-Thickness" problem.

In the case of infinitely thin blades, the presence of each blade is modeled by arranging vortices along the blade camber. In the design problem the swirl schedule $\bar{V}_y'(x)$ is given. It can be shown that the swirl schedule is proportional to the pressure difference (i.e. the loading) accross the blade [Appendix A].

The velocity is divided into two parts: an average velocity $\underset{\sim}{\bar{V}}(x)$, and a perturbed velocity $\underset{\sim}{\tilde{v}}(x,y)$, i.e

$$\underset{\sim}{V}(x,y) \equiv \underset{\sim}{\bar{V}}(x) + \underset{\sim}{\tilde{v}}(x,y) \qquad (2.2\text{-}1)$$

where the average velocity is here taken to be the pitch average defined by:

$$\underset{\sim}{\bar{V}}(x) \equiv \frac{1}{s}\int_{f}^{f+s} \underset{\sim}{V}(x,y)\, dy$$

and thus, by the definition of equation (2.2-1), we must have

$$\int_{f}^{f+s} \underset{\sim}{\tilde{v}}(x,y)\, dy = 0$$

The above equation suggests that if we are to represent $\underset{\sim}{\tilde{v}}$ as a series of smooth functions, perhaps we should choose functions which possess this

zero pitch average property. In chapter 3, we will show that the "smoothing" functions used to represent $\tilde{\underset{\sim}{v}}$ possess this property.

Consider the vorticity field. Given the assumption of incompressible, inviscid and uniform inlet condition, then by Kelvin's theorem, the vortices must lie on the blade camber, and the flow must be irrotational everywhere else.

It can be shown that the vorticity is related to the blade surface α and the y-component of the pitch average velocity $\bar{\underset{\sim}{V}}(x)$ by [Appendix B]:

$$\underset{\sim}{\Omega}(x,y) = s\,\delta_p(\alpha)\left[\nabla\alpha \times \nabla G\right] \tag{2.2-2}$$

where $\delta_p(\alpha)$ is the "periodic delta" function [Appendix B]

Further, $G(x) = -\bar{V}_y(x)$

The pitch average vorticity $\bar{\underset{\sim}{\Omega}}$ is defined as:

$$\begin{aligned}\bar{\underset{\sim}{\Omega}}(x) &\equiv \frac{1}{s}\int_f^{f+s} \underset{\sim}{\Omega}(x,y)\,dy \\ &= \nabla\alpha \times \nabla G\end{aligned} \tag{2.2-3}$$

The vorticity $\underset{\sim}{\Omega}$ is defined as:

$$\underset{\sim}{\Omega}(x,y) \equiv \bar{\underset{\sim}{\Omega}}(x) + \tilde{\underset{\sim}{\Omega}}(x,y)$$

and thus, the perturbed vorticity $\tilde{\underset{\sim}{\Omega}}$ is

$$\begin{aligned}\tilde{\underset{\sim}{\Omega}}(x,y) &= \left[s\,\delta_p(\alpha) - 1\right]\nabla\alpha \times \nabla G \\ &= \nabla S(\alpha) \times \nabla G\end{aligned} \tag{2.2-4}$$

where $S(\alpha)$ is the "sawtooth" function [Appendix B].

We are now in the position to write down the governing equation for the velocity field $\underset{\sim}{V}$. From equation (2.2-4), we can write

$$\nabla \times \tilde{\underset{\sim}{v}} = \nabla S \times \nabla G \tag{2.2-5}$$

The Clebsh formulation says that, to satisfy (2.2-5), write the perturbed velocity $\tilde{\underset{\sim}{v}}$ as the sum of a potential part and a rotational part:

$$\tilde{\underset{\sim}{v}}(x,y) = \nabla \tilde{\phi}(x,y) + S(\alpha)\nabla G(x) \tag{2.2-6}$$

Note that the curl of $\tilde{\underset{\sim}{v}}$ gives $\tilde{\underset{\sim}{\Omega}}$ back.

If the flow is incompressible, then $\nabla . \underset{\sim}{V} = 0 = \nabla . \bar{\underset{\sim}{V}} = \nabla . \tilde{\underset{\sim}{v}}$. Thus, from equation (2.2-6),

$$\begin{aligned} \nabla^2 \tilde{\phi}(x,y) &= -\nabla . (S \nabla G) \\ &= -\nabla S . \nabla G - S \nabla^2 G \end{aligned} \tag{2.2-7}$$

This gives a Poisson equation to be solved for $\tilde{\phi}$ with appropriate boundary conditions (e.g. on the blades).

Finally, the velocity field in the cascade region is of the form:

$$\underset{\sim}{V}(x,y) = \bar{\underset{\sim}{V}}(x) + \left[\nabla \tilde{\phi}(x,y) + S(\alpha)\nabla G(x)\right] \tag{2.2-8}$$

In the design problem, $\bar{\underset{\sim}{V}}$ is given and we seek the solution for $\underset{\sim}{V}$ which satisfies all necessary boundary conditions. In chapter 3, we present a design method for infinitely thin blades using the above theory. In chapter 4, we present a design method for blades having finite thickness based in part on the above approach. In this case, the gap average velocity is given instead of the pitch average velocity, and the above theory needs to be modified.

Chapter 3 : The 'Zero-Thickness' problem

The "Zero-Thickness" problem refers to the design problem of infinitely thin blades. The blades are represented by a distributed bound vorticity which is related to the y-component of the pitch average velocity by equation (2.2-2). We seek the velocity field which must satisfy equation (2.2-8) in the cascade region and appropriate boundary conditions. The blade camber lines are solved iteratively using the blade boundary conditions. The flow is assumed to be incompressible and inviscid. The upstream flow condition is assumed to be uniform.

3.1 Flow regions

The flow field is divided into three regions (figure 1).

(1) Region 1

Region 1 is defined in the interval $-\infty < X < 0$. In this region, the flow is irrotational everywhere. We may write

$$\begin{aligned} \underset{\sim}{V}_1(x,y) &= \underset{\sim}{\bar{V}}(-\infty) + \underset{\sim}{\tilde{v}}_1(x,y) \\ \underset{\sim}{\tilde{v}}_1(x,y) &= \nabla \varphi_{h_1}(x,y) \end{aligned} \tag{3.1-1}$$

where $\underset{\sim}{\bar{V}}(-\infty)$ is the inlet flow condition.

The boundary conditions in this region are:

- far upstream, $\nabla \varphi_{h_1} = 0$
- at the leading edge, $\underset{\sim}{V}_1 = \underset{\sim}{V}_2$

(2) Region 2

Region 2 is defined in the interval $0 < X < 1$. In this region, the flow is rotational and must satisfy equation (2.2-8), i.e.

$$\begin{aligned} \underset{\sim}{V}_2(x,y) &= \overline{\underset{\sim}{V}}(x) + \tilde{\underset{\sim}{v}}_2(x,y) \\ \tilde{\underset{\sim}{v}}_2(x,y) &= \nabla \tilde{\phi}(x,y) + S(x) \nabla G(x) \end{aligned} \tag{3.1-2}$$

where $\overline{\underset{\sim}{V}}(x)$ is the prescribed pitch average velocity.

The boundary conditions in this region are:

- matching conditions at the leading and trailing edges
- no flow-through conditions at the blade surfaces

(3) Region 3

Region 3 is defined in the interval $1 < X < +\infty$. In this region, the flow is irrotational. We may write

$$\begin{aligned} \underset{\sim}{V}_3(x,y) &= \overline{\underset{\sim}{V}}(+\infty) + \tilde{\underset{\sim}{v}}_3(x,y) \\ \tilde{\underset{\sim}{v}}_3(x,y) &= \nabla \varphi_{h_3}(x,y) \end{aligned} \tag{3.1-3}$$

where $\overline{\underset{\sim}{V}}(+\infty)$ is the outlet flow condition.

The boundary condtions in this region are:

- far downstream, $\nabla \varphi_{h3} = 0$
- at the trailing edge, $\underset{\sim}{V}_2 = \underset{\sim}{V}_3$

3.2 Solution of the velocity field

In this section, we present our method of solving the velocity fields in the three regions. At the same time, we try to give reasons for our choice of such a method.

The solution for the perturbed velocities $\underset{\sim}{\tilde{v}}_n'$'s are assumed to be of the form of series of smooth functions: Fourier series and "smoothing" series [Appendix C]. In choosing these series solutions, we note that by our definitions of $\underset{\sim}{V}$, equation (2.2-1), we must satisfy the condition of zero pitch average for $\underset{\sim}{\tilde{v}}$. One way of satisfying this condition is to choose functions in the series having zero pitch average. This is one of the property of "smoothing" functions. In addition, these "smoothing" functions have other properties which are ideal for our applications, i.e.

1. they are periodic in the y-direction. Thus, they can be used to represent the periodicity of the velocity field in the cascade.

2. they are derivatives and integrals of one another. This is a useful property for analysis purpose.

3. they are proportional to δ^n, where n = 1,2,3,... . This property is desired in the case of highly-loaded blade where the solidity is high (or δ is less than 1).

4. they have amplitudes which decrease very fast with increasing order "smoothing" functions. Because of this property, very few terms in the smoothing series are needed to represent a smooth function while high accuracy can still be achieved.

In the case of infinitely thin blades, we expand part of the velocity potential $\tilde{\phi}$ in region 2 as a series of these "smoothing" functions. This is the "smoothing" technique. It turns out that by using this technique, the expressions for the velocity fields are very simple to compute.

Refer to equation (3.1-2), we assume the velocity potential $\tilde{\phi}$ in region 2 of the form

$$\tilde{\phi}(x,y) = \tilde{\phi}_s(x,y) + \varphi_{h2}(x,y) \tag{3.2-1}$$

where $\tilde{\phi}_s(x,y) = A(x)\,I(\alpha) + B(x)\,J(\alpha) + C(x)\,K(\alpha) + \ldots$

The coefficients $A(x)$, $B(x)$, $C(x)$, ... are chosen such that the blade boundary conditions are satisfied. This form gives $\tilde{\phi}_s$ to have zero pitch average and curl free.

The "homogeneous part" of the velocity potential φ_{h2} is chosen so as to satisfy $\nabla^2 \varphi_{h2} = 0$, to have zero pitch average, and to match the velocities at the leading and trailing edges with those in regions 1 and 3.

The velocity field in the cascade region can then be written as

$$\underset{\sim}{V}_2(x,y) = \underset{\sim}{\bar{V}}(x) + \left[S(\alpha)\nabla G(x) + \nabla\tilde{\phi}_s(x,y) + \nabla\varphi_{h2}(x,y) \right] \tag{3.2-2}$$

3.2.1 Smoothing series

For incompressible, continuity requires

$$\nabla \cdot \underset{\sim}{V} = 0 \tag{3.2-3}$$

but $\overline{V}_x = 1$ (by continuity of the pitch average flow), and equation (3.2-3) implies

$$\nabla \cdot \tilde{\underset{\sim}{v}} = 0 \tag{3.2-4}$$

or $$\nabla^2 \tilde{\phi}_s = -\nabla \cdot (s \nabla G)$$

Now, from equation (3.2-1), we can write

$$\begin{aligned}
\nabla^2 \tilde{\phi}_s = & -A|\nabla\alpha|^2 + s(\alpha)\left[A\nabla^2\alpha + 2\nabla A \cdot \nabla\alpha + B|\nabla\alpha|^2\right] \\
& + I(\alpha)\left[\nabla^2 A + B\nabla^2\alpha + 2\nabla B \cdot \nabla\alpha + C|\nabla\alpha|^2\right] \\
& + J(\alpha)\left[\nabla^2 B + C\nabla^2\alpha + 2\nabla C \cdot \nabla\alpha + D|\nabla\alpha|^2\right] \\
& \vdots
\end{aligned}$$

To satisfy (3.2-4), we choose

$$\begin{aligned}
A|\nabla\alpha|^2 &= -\nabla\alpha \cdot \nabla G \\
B|\nabla\alpha|^2 &= -\left(\nabla^2 G + 2\nabla A \cdot \nabla\alpha + A|\nabla\alpha|^2\right) \\
C|\nabla\alpha|^2 &= -\left(\nabla^2 A + 2\nabla B \cdot \nabla\alpha + B|\nabla\alpha|^2\right) \\
&\vdots
\end{aligned} \tag{3.2-5}$$

and the pattern for all the coefficients is apparent.

3.2.2 Matching conditions

As mentioned earlier, the velocity potentials φ_{hn}'s are used solely to satisfy the far upstream and far downstream flow conditions, and the matching conditions of the velocities at the leading and trailing edges. We have chosen φ_{hn}'s to satisfy Laplace equation: therefore

we assume φ_{hn}'s of the forms:

$$\varphi_{h1}(x,y) = \sum_{n=1}^{\infty} A_n \, e^{-\lambda_n |x|} \, e^{i\lambda_n y}$$

$$\varphi_{h2}(x,y) = \sum_{n=1}^{\infty} \left(B_n^{+} e^{-\lambda_n |x|} + B_n^{-} e^{-\lambda_n |x-1|} \right) e^{i\lambda_n y}$$

$$\varphi_{h3}(x,y) = \sum_{n=1}^{\infty} C_n \, e^{-\lambda_n |x-1|} \, e^{i\lambda_n y}$$

These assumed forms for φ_{hn}'s satisfy

- Laplace equation
- The far upstream and far downstream flow conditions
- The periodicity condition in the y-direction of the flow field

The unknowns coefficients, A_n, B_n^{+}, B_n^{-} and C_n, are determined using the matching conditions of the velocities at the leading and trailing edges:

$$
\begin{aligned}
V_{1x}(x=0, y) &= V_{2x}(x=0, y) \\
V_{1y}(x=0, y) &= V_{2y}(x=0, y) \\
V_{2x}(x=1, y) &= V_{3x}(x=1, y) \\
V_{2y}(x=1, y) &= V_{3y}(x=1, y)
\end{aligned}
\tag{3.2-6}
$$

Now, the velocities in the three regions are the following.

For region 1,

$$
\underset{\sim}{V}_1 = \underset{\sim}{\bar{V}}(-\infty) + \nabla \varphi_{h_1}
$$

For region 2,

$$
\begin{aligned}
\underset{\sim}{V}_2 = \underset{\sim}{\bar{V}}(x) &+ \nabla \varphi_{h_2} \\
&+ S(\alpha)(\nabla G + A \nabla \alpha) \\
&+ I(\alpha)(\nabla A + B \nabla \alpha) \\
&+ J(\alpha)(\nabla B + C \nabla \alpha) \\
&\vdots
\end{aligned}
\tag{3.2-7}
$$

For region 3,

$$
\underset{\sim}{V}_3 = \underset{\sim}{\bar{V}}(+\infty) + \nabla \varphi_{h_3}
$$

The "sawtooth" function $S(\alpha)$ has a finite jump across any constant α lines. At the leading and trailing edges, we assume that the flow comes and leaves smoothly (zero-incidence and Kutta conditions respectively). Thus, from (3.2-7), we require at $X = 0$ and $X = 1$,

$$(\nabla G + A \nabla \alpha) = 0$$

or

$$\begin{aligned} A(X = 0, 1) &= 0 \\ G'(X = 0, 1) &= 0 \end{aligned} \tag{3.2-8}$$

From equation (3.2-5), by choosing a loading distribution with the condition $G'(X = 0, 1) = 0$, we automatically satisfy the condition $A(X = 0, 1) = 0$.

Substituting equation (3.2-7) into equation (3.2-6) using equation (3.2-8), we can show that, in region 2

$$\begin{aligned} \varphi_{h_2}(x,y) = \sum_{t=0}^{1} \sum_{n=1}^{\infty} \frac{e^{-\lambda_n |x-t|}}{\lambda_n^3} \Bigg\{ & \\ & + \left(B(t) \sin \lambda_n [y - f(t)] - (-1)^t [A'(t) - B(t) f'(t)] \cos \lambda_n [y - f(t)] \right) \\ & - \frac{1}{\lambda_n} \left(C(t) \cos \lambda_n [y - f(t)] - (-1)^t [B'(t) - C(t) f'(t)] \sin \lambda_n [y - f(t)] \right) \\ & \cdots \Bigg\} \end{aligned} \tag{3.2-9}$$

where

$$\lambda_n = \frac{2\pi n}{s}$$

3.2.3 Blade boundary conditions

The blade boundary conditions are used to generate the blade camber. By adding the blade boundary conditions together, we obtain

$$\langle \underset{\sim}{V} \rangle \cdot \nabla \alpha = 0 \qquad (3.2\text{-}10)$$

Define

$$f_o' = \frac{\bar{V}_y}{\bar{V}_x}$$

Then, f_o represents the meam streamline.

If we define the blade camber line as

$$f \equiv f_o + \Delta f$$

then, we can show that equation (3.2-10) is a perfect derivative in Δf and reduces to

$$\begin{aligned}\Delta f(x) = & -I(0)(G' - Af') - K(0)(B' - Cf') - \cdots \\ & - \sum_{t=0}^{1} \sum_{n=1}^{\infty} \frac{e^{-\lambda_n |x-t|}}{\lambda_n^3} \Big\{ \\ & \left[(-1)^t B(t) - \frac{1}{\lambda_n}\left[B'(t) - C(t)f'(t)\right] \cdots \right] \cos\lambda_n \left[f(x) - f(t)\right] \\ & + \left[\left[A'(t) - B(t)f'(t)\right] + \frac{(-1)^t}{\lambda_n} C(t) \cdots \right] \sin\lambda_n \left[f(x) - f(t)\right] \Big\} \end{aligned} \qquad (3.2\text{-}11)$$

3.3 Iteration process for the blade camber line

We are now in the position to compute the blade shape using equation (3.2-11). We call this equation the "camber generator". The beauty of the

"smoothing" technique is that the blade shape and the velocity field can be obtained by solving a set of algebriac equations, i.e. equations (3.2-5) and (3.2-11):

$$\left.\begin{aligned} A(x) &= \frac{f'G'}{(1+f'^2)} \\ B(x) &= \frac{-G'' + 2A'f' + Af''}{(1+f'^2)} \\ C(x) &= \frac{-A'' + 2B'f' + Bf''}{(1+f'^2)} \\ &\vdots \end{aligned}\right\} \qquad (3.3\text{-}1)$$

$$f(x) = f_0(x) + \Delta f(x) \qquad (3.3\text{-}2)$$

Thus, we have (n+1) equations and (n+1) unknowns. The (n+1) unknowns are: n coefficients in the smoothing series : $A(x)$, $B(x)$, $C(x)$ ... , plus the blade camber line $f(x)$.

We choose to solve the blade camber line by an iteration process. We expect that the blade camber line f is not very different from the mean streamline f_0 , especially in the case where the spacing s is small. Thus we can use f_0 as the initial guess for f and go through the following iteration process:

1. Compute A , B , C , ... (with the guessing value of f as f_0 at the first iteration) using equations (3.3-1)

2. Update f by solving equation (3.3-2)

3. Go back to step 1 to update A , B , C , ... with the new value of f computed in step 2. Continue this process until convergence in f is achieved.

The whole flow field and the pressure coefficients at the blade surfaces can be computed using equations (3.2-7) and Bernoulli's equations.

3.4 "Partial smoothing"

The proposed iteration process for the camber line in section 3.3 shows that we can get infinite accuracy for f by keeping an infinite number of terms in the smoothing series. From the engineering point of view, however, we want to take the least number of terms in the smoothing series for a given required accuracy criteria. "Partial smoothing" refers to the truncation of the smoothing series.

In the inverse poblem, we are most interested in three quantities: the blade camber line and the pressure coefficients at the blade upper and lower surfaces. Since $f' \sim V$, we see that f has the same order of accuracy as ϕ . Similarly, since $C_p \sim V^2$, we see that C_p also has the same order of accuracy as ϕ . We now construct a table showing the truncation errors in f and C_p as a function of the number of terms used in the smoothing series $\tilde{\phi}$.

Number of terms kept in the smoothing series	Truncation errors
2	$K(0)$
4	$M(0)$
⋮	⋮

3.5 Numerical method

As an example of the design problem, we write a program to compute the blade camber line by keeping two terms in the smoothing series. In this case, the smoothing series $\tilde{\phi}$ is of the form:

$$\tilde{\phi}(x,y) = A(x)\,I(\alpha) + B(x)\,J(\alpha)$$

Let's investigate the order of magnitude of the truncation error in f when the above "partial smoothing" form is used. From equation (3.2-11), we see that the truncation error in f is of the order of $K(0)$, or [Appendix C]:

$$\text{error in } f(x) \sim \frac{\delta^4}{720}$$

Therefore, for the above example, the truncation error in f is about three orders of magnitude less than δ^4. By keeping only two terms in the smoothing series, we can still maintain very high accuracy (accurate to $.0014\,\delta^4$). Moreover, the numerical calculation is very simple. Given the loading distribution $\bar{V}_y'(x)$ as an analytical form, the blade camber can be calculated through a set of algebriac equation. No numerical technique is needed to compute derivatives or integrals.

The iteration process for f involves calculating the following parameters:

1. Guess f and its derivatives as:

$$f(x) = f_o(x) + \frac{s^2}{12}(G' - Af_o')$$

$$f'(x) = f_o'(x)$$

$$f''(x) = f_o''(x)$$

for simplicity, we set $f''(x) = f_o''(x)$ through out the iteration process

2. Compute

$$A(x) = \frac{f'G'}{(1+f'^2)}$$

$$A'(x) = \frac{(1+f'^2)(f'G'' + f''G') - 2f'^2 f''G'}{(1+f'^2)^2}$$

$$B(x) = \frac{-G'' + 2f'A' + Af''}{(1+f'^2)}$$

3. Update

$$f'(x) = f_o'(x) + \frac{s^2}{12}(G'' - A'f' - Af'') - \frac{d}{dx}(FSUM)$$

$$f(x) = f_o(x) + \frac{\delta^2}{12}(G' - Af') - FSUM$$

where

$$FSUM = \sum_{t=0}^{1}\sum_{n=1}^{\infty} \frac{e^{-\lambda_n|x-t|}}{\lambda_n^3}\left\{(-1)^t B(t)\cos\lambda_n[f(x)-f(t)] + [A'(t)-B(t)f'(t)]\sin\lambda_n[f(x)-f(t)]\right\}$$

4. Go back to step 2 and repeat the calculations until convergence in f is achieved.

The convergence criteria used in the calculation is, at each location X_i,

$$f^{n+1}(x_i) - f^{n}(x_i) \leqslant \text{ERROR}$$

i.e. if the difference between the present value of f and the previous value of f at all locations X_i is less than ERROR (the convergence criteria), then convergence in f is achieved.

3.4 Numerical results

Numerical results shows that comvergence in f can be achieved fast, depending on the spacing to chord ratio δ. In this section, we present a numerical example. The results are compared with the "exact" solution (using the Biot-Savart law) [4].

We have taken the case of an inlet guide vane with the following inputs:

- spacing to chord ratio $s = 0.75$
- inlet angle $\alpha_1 = 0^\circ$
- outlet angle $\alpha_2 = 45^\circ$
- loading distribution $\Delta P \sim X(1-X)$

Calculations are made at 21 points and the convergence criteria ERROR is 10^{-5}.

17 iterations are required for f to converge. The computer program output is shown in table 1. Figure 2 show the plots of the blade camber line and the mean streamline. Note that there is reverse curvature of the blade near the trailing edge. This observation is also found in the results performed by Tan, Wang, Hawthorne and McCune [4]. The computational time for this example is around 7 seconds CPU time on the Digital VAX-11 computer.

Table 2 shows a comparason of the blade camber obtained from the "smoothing" technique and the "exact" method. The results compare well.

This preliminary study of the "smoothing" technique shows that it is indeed a very powerful engineering tool for the design problem in terms of computational time and accuracy (if desired).

Chapter 4 : The "Finite-Thickness" Problem

In this chapter, we will present a method to solve the inverse problem for blades having finite thickness. The method will be similar to the one presented in the "Zero-Thickness" problem, except that we now have to differentiate the average flow quantities. In the case of blades having finite thickness, we can define two average quantities:

1. The pitch average velocity which is defined as:

$$\underset{\sim}{\overline{V}}(x) \equiv \frac{1}{s}\int_{f}^{f+s} \underset{\sim}{V}(x,y)\, dy$$

The pitch average velocity has no physical representation of the flow since there is no flow in the blade. However, if we are to model the presence of the blades by distributing singularities on the blade camber lines, then $\underset{\sim}{\overline{V}}$ does exist in this sense.

2. The gap average velocity which is defined as:

$$\underset{\sim}{\overline{V}}_T(x) \equiv \frac{1}{s - 2T(x)} \int_{f+T}^{f+s-T} \underset{\sim}{V}(x,y)\, dy$$

Obviously, the gap average velocity does represent the average of the actual flow.

In the design problem of blades having finite thickness, as mentioned in the introduction, we choose to model the presence of the blades by distributing vortices, sources and sinks on the blade camber lines. In our method, the two main given quantities are: the loading distribution $\overline{V}'_{Ty}$, and the blockage distribution T.

Given these two quantities, the gap average velocity is known. It is defined by:

$$\underset{\sim}{\bar{V}}_T(x) = \bar{V}_{Tx}(x)\,\hat{e}_x + \bar{V}_{Ty}(x)\,\hat{e}_y$$

where $\bar{V}_{Tx}(x)$ is found using continuity, i.e.

$$\bar{V}_{Tx}(x) = \frac{s}{s - 2T(x)}$$

By distributing vortices on the blade camber lines, we have shown that the strength of the vorticity is related to the y-component of the pitch average velocity through equation (2.2-2). Therefore, in the "Finite-Thickness" problem, we no longer know the strength of the vorticity. Equation (2.2-8) is still valid in describing the flow field in region 2, however, we now choose a different way to describe it.

4.1 Flow regions

Using the same general approach as in the "Zero-Thickness" problem, we again divide the flow field into three regions. The velocity potentials ψ_{hn}'s are used to satisfy the far upstream and far downstream flow conditions, and the matching conditions of the velocities at the leading and trailing edges. We again make use of the "smoothing" functions to satisfy the blade boundary conditions.

The three flow regions are:

(1) Region 1

Region 1 is defined in the interval $-\infty < X < 0$. In this region, the flow is irrotational everywhere. We may write

$$\underset{\sim}{V}_1(x,y) = \underset{\sim}{\bar{V}}(-\infty) + \underset{\sim}{\tilde{v}}_1(x,y)$$

$$\underset{\sim}{\tilde{v}}_1(x,y) = \nabla \varphi_{h1}(x,y) \qquad (4.1\text{-}1)$$

where $\underset{\sim}{\bar{V}}(-\infty)$ is the inlet flow condition. The boundary conditions are:

- far upstream $\nabla \varphi_{h1} = 0$
- at the leading edge $\underset{\sim}{V}_1 = \underset{\sim}{V}_2$

(2) Region 2

Region 2 is defined in the interval $0 < X < 1$. We choose to analyze the flow in a region between the blade camber lines. The flow in this region is divergence free and curl free. We may write

$$\underset{\sim}{V}_2(x,y) = \nabla \phi(x,y)$$

$$\nabla^2 \phi = 0 \qquad (4.1\text{-}2)$$

The boundary conditions in this region are:

- matching conditions at the leading and trailing edges
- no flow-through condition at the blade surfaces

(3) Region 3

Region 3 is defined in the interval $1 < X < +\infty$. In this region, the flow is irrotational. We may write

$$\underset{\sim}{V}_3(x,y) = \underset{\sim}{\bar{V}}(+\infty) + \underset{\sim}{\tilde{v}}_3(x,y)$$

$$\underset{\sim}{\tilde{v}}_3(x,y) = \nabla \varphi_{h3}(x,y) \qquad (4.1\text{-}3)$$

where $\underset{\sim}{\bar{V}}(+\infty)$ is the oulet flow condition. The boundary conditions in this region are:

- far downstream $\nabla \varphi_{h3} = 0$
- at the trailing edge $\underset{\sim}{V}_2 = \underset{\sim}{V}_3$

4.2 Solution of the velocity field

Consider the flow between the two camber lines (excluding these camber lines) located at $y = f^+$ and $y = f^- + \delta$. In this region, we can write $\underset{\sim}{V}_2$ as a velocity potential satisfying Laplace equation:

$$\underset{\sim}{V}_2(x,y) = \nabla \phi(x,y)$$

$$\nabla^2 \phi = 0 \tag{4.2-1}$$

Let's assume ϕ to be of the form

$$\phi(x,y) = S(\alpha)\, G_T(x) + \bar{\phi}(x) + \phi_{TS}(x,y) + \varphi_{h2}(x,y) \tag{4.2-2}$$

where the smoothing series ϕ_{TS} is assumed to be of the form

$$\phi_{TS}(x,y) = \bar{\phi}_{TS}(x) + \delta(x)\, S(\alpha) + A_T(x)\, I(\alpha) + B_T\, J(\alpha) + \ldots \tag{4.2-3}$$

and thus, ϕ can be written as

$$\phi(x,y) = \left[G_T + \delta\right] S(\alpha) + \bar{\phi} + \bar{\phi}_{TS} + \varphi_{h2} \quad A_T\, I(\alpha) + B_T\, J(\alpha) + \ldots \tag{4.2-4}$$

Combining equation (4.2-1) , (4.2-2) along with the choice $\nabla^2 \varphi_{h2} = 0$, we obtain

$$\nabla^2 \phi_{TS} = -\nabla^2 \bar{\phi} - S \nabla^2 G_T + 2 \nabla G_T \cdot \nabla \alpha - G_T \nabla^2 S \tag{4.2-5}$$

Based on the experience from the "Zero-Thickness" problem, there are a few motivations in choosing ϕ of the above form:

- δ in the smoothing series ϕ_{TS} is the additional amount of vorticity needed to represent the total amount of vorticity distributed on the blade camber lines.
- $\bar{\phi}$ and $\bar{\phi}_{TS}$ represent some average in the potential velocity which we have the freedom to choose at our convenience.
- ϕ_{h2} is chosen to satisfy the matching conditions at the leading and trailing edges.
- A_T , B_T , C_T , ... are chosen to satisfy the blade boundary conditions.

We are now in the position to solve for the velocity field in region 2. Our tasks are to:

1. relate some of the unknowns in the assumed form of ϕ to the given gap average velocity $\bar{\underset{\sim}{V}}_T$
2. satisfy the Poisson equation (4.2-5)
3. satisfy the blade boundary conditions

$$\underset{\sim}{V}_2 \cdot \nabla(\kappa - T) = 0 \qquad (4.2\text{-}6)$$

$$\underset{\sim}{V}_2 \cdot \nabla(\kappa + T) = 0 \qquad (4.2\text{-}7)$$

4.2.1 Relation between the gap average velocity and the unknowns

In this subsection, we relate some of the unknowns in the velocity potential ϕ defined in equation (4.2-4) to the gap average velocity.

By definition of the gap average velocity,

$$\int_{f+T}^{f+s-T} \underset{\sim}{V}_2 \, dy = (s - 2T)\,\overline{\underset{\sim}{V}}_T$$

$$= 2\,S(T)\,\overline{\underset{\sim}{V}}_T$$

Evaluating the above integral using equation (4.2-4), we obtain

$$(\bar{\phi}' + f' G_T)\,\hat{e}_x + (-G_T)\,\hat{e}_y + \overline{(\nabla\phi_{TS} + \nabla\psi_{h2})}_T = \overline{\underset{\sim}{V}}_T$$

We now choose the coefficients in ϕ such that

$$(\bar{\phi}' + f' G_T)\,\hat{e}_x + (-G_T)\,\hat{e}_y = \overline{\underset{\sim}{V}}_T$$

Therefore, we require

$$(\bar{\phi}' + f' G_T) = \bar{V}_{Tx} \tag{4.2-8}$$

and

$$(\nabla\phi_{TS} + \nabla\psi_{h2})_T = 0 \tag{4.2.9}$$

Equation (4.2-8) requires

$$\bar{\phi}' = \bar{V}_{Tx} - f' G_T \tag{4.2-10}$$

Equation (4.2-9), a vector quantity, can be satisfied by choosing

$$\delta = -\frac{J(T)}{S(T)} B_T - \frac{L(T)}{S(T)} D_T - \cdots + \overline{\left(\frac{\partial\psi_{h2}}{\partial y}\right)}_T \tag{4.2-11}$$

and

$$\bar{\phi}'_{TS} = \frac{J(T)}{S(T)}(A'_T - B_T f') + \frac{L(T)}{S(T)}(C'_T - D_T f') \cdots - \delta f' - \overline{\left(\frac{\partial\psi_{h2}}{\partial x}\right)}_T \tag{4.2-12}$$

Define

$$\beta = \frac{J(\tau)}{S(\tau)}\left(A'_\tau - B_\tau f'\right) + \frac{L(\tau)}{S(\tau)}\left(C'_\tau - D_\tau f'\right) + \cdots \qquad (4.2\text{-}13)$$

$$-\left(\overline{\frac{\partial \varphi_{h2}}{\partial x}}\right)_\tau$$

We will show later that β relates to the source/sink strength.

4.2.2 Satisfaction of the Poisson equation

We now choose the coefficients in the smoothing series so that the Poisson equation (4.2-5) can be satisfied.

From the assumed form of the smoothing series ϕ_{TS}, equation (4.2-3), we can write

$$\begin{aligned}\nabla^2\phi_{TS} = {} & \left(\nabla^2\bar{\phi}_{TS} - \delta\nabla^2\alpha - 2\nabla\delta\cdot\nabla\alpha - A_\tau|\nabla\alpha|^2\right) \\ & + S(\alpha)\left(\nabla^2\delta + A_\tau\nabla^2\alpha + 2\nabla A_\tau\cdot\nabla\alpha + B_\tau|\nabla\alpha|^2\right) \\ & + I(\alpha)\left(\nabla^2 A_\tau + B_\tau\nabla^2\alpha + 2\nabla B_\tau\cdot\nabla\alpha + C_\tau|\nabla\alpha|^2\right) \\ & + J(\alpha)\left(\nabla^2 B_\tau + C_\tau\nabla^2\alpha + 2\nabla C_\tau\cdot\nabla\alpha + D_\tau|\nabla\alpha|^2\right) \\ & + \cdots\end{aligned}$$

Substituting the above equation into equation (4.2-5) using equations (4.2-10) through (4.2-12), we obtain, in scalar form

$$A_T |\nabla\alpha|^2 - (\bar{V}'_{Tx} + \beta') - f'(G'_T + \delta')$$
$$= S(\alpha)\left[(G''_T + \delta'') - A_T f'' - 2A'_T f' + B_T |\nabla\alpha|^2\right]$$
$$+ I(\alpha)\left[A''_T - B_T f'' - 2B'_T f' + C_T |\nabla\alpha|^2\right]$$
$$+ J(\alpha)\left[B''_T - C_T f'' - 2C'_T f' + D_T |\nabla\alpha|^2\right]$$
$$+ \cdots$$

The above equation is of the form

$$a_0(x) = g_1(x,y)\, a_1(x) + g_2(x,y)\, a_2(x) + \cdots$$

One way to satisfy the above equation is to choose

$$a_0(x) = 0$$
$$a_1(x) = 0$$
$$a_2(x) = 0$$
$$\vdots$$

Using the choices suggested above, we obtain

$$A_T |\nabla\alpha|^2 = f'(G'_T + \delta') + (\bar{V}'_{Tx} + \beta')$$

$$
\begin{aligned}
B_T\,|\nabla\alpha|^2 &= -\left(G_T'' + \delta''\right) + A_T f'' + 2A_T' f' \\
C_T\,|\nabla\alpha|^2 &= -A_T'' + B_T f'' + 2B_T' f' \\
D_T\,|\nabla\alpha|^2 &= -B_T'' + C_T f'' + 2C_T' f' \\
&\vdots
\end{aligned}
\tag{4.2-14}
$$

By choosing A_T, B_T, C_T, ... of the above forms, it turns out that one of the blade boundary conditions is satisfied. This fact will be discussed in section (4.2.4).

4.2.3 Matching conditions

As in the infinitely thin blade case, the velocity potential φ_{hn}'s are used to satisfy the matching conditions of the velocities at the leading and trailing edges, and the far upstream and far downstream flow conditions. Again, φ_{hn}'s are chosen to satisfy Laplace equation. A similar analysis for φ_{h2} as in the "Zero-Thickness" problem gives:

$$
\begin{aligned}
\varphi_{h2}(x,y) = \sum_{t=0}^{1}\sum_{n=1}^{\infty} \frac{e^{-\lambda_n|x-t|}}{\lambda_n^3} \Big\{ & \\
& + \Big(B_T(t)\sin\lambda_n\left[y-f(t)\right] - (-1)^t\left[A_T'(t) - B_T(t)f'(t)\right]\cos\lambda_n\left[y-f(t)\right]\Big) \\
& - \frac{1}{\lambda_n}\Big(C_T(t)\cos\lambda_n\left[y-f(t)\right] - (-1)^t\left[B_T'(t) - C_T(t)f'(t)\right]\sin\lambda_n\left[y-f(t)\right]\Big) \\
& + \quad \cdots \quad \Big\}
\end{aligned}
\tag{4.2-15}
$$

where $\lambda_n = \frac{2\pi n}{s}$

With the analysis done so far, we can now write down the velocity in region 2:

$$\begin{aligned} \underset{\sim}{V}_2(x,y) = \underset{\sim}{\bar{V}}_T(x) &+ \left[\beta(x)\hat{e}_x - \delta(x)\hat{e}_y\right] + \nabla\phi_{b2}(x,y) \\ &+ S(\alpha)\left[\nabla\left(G_T(x)+\delta(x)\right) + A_T(x)\nabla\alpha\right] \\ &+ I(\alpha)\left[\nabla A_T(x) + B_T(x)\nabla\alpha\right] \\ &+ J(\alpha)\left[\nabla B_T(x) + C_T(x)\nabla\alpha\right] + \cdots \end{aligned} \tag{4.2-16}$$

Again, we require

$$\left.\begin{aligned} A_T &= 0 \\ G_T' + \delta' &= 0 \end{aligned}\right\} \quad \text{at} \quad X = 0,1$$

for the flow to come and leave smoothly leading and trailing edges (the zero-incidence and the Kutta conditions respectively). Using the definition of A_T in equation (4.2-14), the above conditions become:

$$\left.\begin{aligned} G_T' + \delta' &= 0 \\ \bar{V}_{Tx}' + \beta' &= 0 \end{aligned}\right\} \quad \text{at} \quad X = 0,1 \tag{4.2-17}$$

4.2.4 Blade boundary conditions

So far, we have chosen all the coefficients in the expression for ϕ. We must now satisfy the two blade boundary conditions, i.e. equations (4.2-6) and (4.2-7). It turns out that one boundary condition is automatically satisfied

through the choices of A_T , B_T , C_T , ... in equation (4.2-14) while the other blade boundary condition is used to generate the blade camber line.

By adding and subtracting equations (4.2-6) and (4.2-7), the blade boundary conditions become:

$$< \nabla\phi >_T \cdot \nabla\alpha \; - \; \Delta_T (\nabla\phi) \cdot \nabla T \; = \; 0$$

$$\Delta_T (\nabla\phi) \cdot \nabla\alpha \; - \; < \nabla\phi >_T \cdot \nabla T \; = \; 0$$

Evaluating $\nabla\phi$ at the upper and lower sufaces of the blade and substituting them into the above equations, we obtain:

$$\begin{aligned} &-(G_T+\delta) + (\nabla\bar{\phi} + \nabla\bar{\phi}_{Ts}) \cdot \nabla\alpha + < \nabla\psi_{h2} >_T \cdot \nabla\alpha \\ &+ I(T) \left[\nabla A_T \cdot \nabla\alpha + B_T |\nabla\alpha|^2 \right] \\ &+ K(T) \left[\nabla C_T \cdot \nabla\alpha + D_T |\nabla\alpha|^2 \right] + \cdots \\ &\qquad = \Delta_T (\nabla\psi_{h2}) \cdot \nabla T \\ &\qquad\quad + S(T) \left[\nabla(G_T+\delta) \cdot \nabla T + A_T \nabla\alpha \cdot \nabla T \right] \\ &\qquad\quad + J(T) \left[\nabla B_T \cdot \nabla T + C_T \nabla\alpha \cdot \nabla T \right] + \cdots \end{aligned} \tag{4.2-18}$$

and

$$
\begin{aligned}
&\Delta_T(\nabla\psi_{h2})\cdot\nabla\alpha \\
&+ S(T)\left[\nabla(G_T+\delta)\cdot\nabla\alpha + A_T|\nabla\alpha|^2\right] \\
&+ J(T)\left[\nabla B_T\cdot\nabla\alpha + C_T|\nabla\alpha|^2\right] + \cdots \\
&\quad = -(G_T+\delta)\nabla\alpha\cdot\nabla T + (\nabla\bar{\phi}+\nabla\bar{\phi}_{Ts})\cdot\nabla T \\
&\qquad + \langle\nabla\psi_{h2}\rangle_T\cdot\nabla T \\
&\qquad + I(T)\left[\nabla A_T\cdot\nabla T + B_T\nabla\alpha\cdot\nabla T\right] \qquad (4.2\text{-}19) \\
&\qquad + K(T)\left[\nabla C_T\cdot\nabla T + D_T\nabla\alpha\cdot\nabla T\right] + \cdots
\end{aligned}
$$

By substituting the choices of A_T, B_T, C_T ... in equations (4.2-14) into equation (4.2-19) and along with the identity

$$
\frac{d}{dX}\left[S(T)\left(\overline{\frac{\partial\psi_{h2}}{\partial x}}\right)_T\right] = \Delta_T(\nabla\psi_{h2})\cdot\nabla\alpha - \langle\nabla\psi_{h2}\rangle_T\cdot\nabla T
$$

we can show that the blade boundary condition (4.2-19) is automatically satisfied.

Finally, the other blade boundary condition, equation (4.2-18), is satisfied by choosing the blade camber line f appropriately. We will call equation (4.2-18) the "camber generator".

Substituting the choices of the coefficients A_T, B_T, C_T, ... into the blade boundary conditions (4.2-18), we obtain

$$S(T)\,\bar{\underset{\sim}{V}}_T \cdot \nabla\alpha = S(T)\overline{(\nabla\psi_{hL})}_T \cdot \nabla\alpha + S(T)\left\{\Delta_T(\nabla\psi_{hL})\cdot\nabla T - \langle \nabla\psi_{hL}\rangle_T \cdot \nabla\alpha\right\} + \left[J(T)\frac{d}{dx} + S(T)\frac{d}{dx}I(T)\right]\left\{(G_T' + \delta') - A_T f'\right\} + \left[L(T)\frac{d}{dx} + S(T)\frac{d}{dx}K(T)\right]\left\{B_T' - C_T f'\right\} + \cdots \qquad (4.2\text{-}20)$$

This is a differential equation relating f and the coefficients A_T, B_T, C_T, ... As in the "Zero-Thickness" problem, we can reduce the above equation into an algebriac equation for f (in the iterative sense).

Again, define

$$f_o'(x) \equiv \frac{\bar{V}_{Ty}}{\bar{V}_{Tx}}$$

Then f_o represents the mean streamline.

Also, define

$$f(x) = f_o(x) + \Delta f(x)$$

Then, referring to equation (4.2-20), it can be shown that

$$S(T)\,\bar{\underset{\sim}{V}}_T \cdot \nabla\alpha = -\frac{s}{2}\frac{d}{dx}(\Delta f)$$

$$\left[J(\tau) \frac{d}{dx} + S(\tau) \frac{d}{dx} I(\tau) \right] \left\{ (G_\tau' + \delta') - A_\tau f' \right\}$$
$$= \frac{d}{dx} \left\{ \left[J(\tau) + S(\tau) I(\tau) \right] \left[(G_\tau' + \delta') - A_\tau f' \right] \right\}$$

$$\left[L(\tau) \frac{d}{dx} + S(\tau) \frac{d}{dx} K(\tau) \right] \left\{ B_\tau' - C_\tau f' \right\}$$
$$= \frac{d}{dx} \left\{ \left[L(\tau) + S(\tau) K(\tau) \right] \left[B_\tau' - C_\tau f' \right] \right.$$
$$\vdots$$

and finally

$$S(\tau) \left\{ \Delta_\tau (\nabla \psi_{hz}) \cdot \nabla T - \langle \nabla \psi_{hz} \rangle_\tau \cdot \nabla \alpha \right\}$$
$$+ S(\tau) \left(\overline{\nabla \psi_{hz}} \right)_\tau \cdot \nabla T = \frac{d}{dx} (FSUM)$$

where

$$FSUM = \sum_{t=0}^{1} \sum_{n=1}^{\infty} \frac{e^{-\lambda_n |x-t|}}{\lambda_n^3} \left(\frac{\sin \lambda_n T}{\lambda_n} + S(\tau) \cos \lambda_n T \right) \left\{ \right.$$
$$\left[(-1)^t B_\tau(t) - \frac{1}{\lambda_n} \left[B_\tau'(t) - C_\tau(t) f'(t) \right] \right.$$
$$\left. - \frac{(-1)^t}{\lambda_n^2} D_\tau(t) + \cdots \right] \cos \lambda_n \left[f(x) - f(t) \right]$$
$$\left[\left[A_\tau'(t) - B_\tau(t) f'(t) \right] + \frac{(-1)^t}{\lambda_n} C_\tau(t) \right.$$
$$\left. \left. - \frac{1}{\lambda_n^2} \left[C_\tau'(t) - D_\tau(t) f'(t) \right] - \cdots \right] \sin \lambda_n \left[f(x) - f(t) \right] \right\}$$

Combining these equations together, the "camber generator" becomes

$$\Delta f(x) = -\frac{2}{s}\left[J(\tau) + S(\tau)\, I(\tau)\right](G_\tau' + \delta' - A_\tau f') - \frac{2}{s}\left[L(\tau) + S(\tau)\, K(\tau)\right](B_\tau' - C_\tau f') - \cdots - \frac{2}{s}(FSUM) \tag{4.2-21}$$

4.3 Iteration process for the blade camber line

We choose a very similar iteration scheme used in the "Zero-Thickness" problem. The iteration process consists of calculating the coefficients A_τ, B_τ, C_τ, ... using equations (4.2-14). Then, update δ' and β' using equations (4.2-11) and (4.2-13). Finally, update f using equation (4.2-21).

In quantitative forms,

1. Compute

$$A_\tau(x) = \frac{f'(G_\tau' + \delta') + (\bar{V}_{\tau x}' + \beta')}{(1 + f'^2)}$$

$$B_\tau(x) = \frac{-(G_\tau'' + \delta'') + A_\tau f'' + 2A_\tau' f'}{(1 + f'^2)} \tag{4.3-1}$$

$$C_\tau(x) = \frac{-A_\tau'' + B_\tau f'' + 2B_\tau' f'}{(1 + f'^2)}$$

$$\mathcal{D}_T(x) = \frac{-B_T'' + C_T f'' + 2C_T' f'}{(1+f'^2)}$$

$$\vdots$$

2. Update

$$\beta' = \frac{d}{dx}\left[\frac{J(T)}{S(T)}(A_T' - B_T f') + \frac{L(T)}{S(T)}(C_T' - \mathcal{D}_T f') + \cdots + \left(\overline{\frac{\partial \varphi_{h_2}}{\partial x}}\right)_T\right]$$

$$\delta' = \frac{d}{dx}\left[-\frac{J(T)}{S(T)} B_T - \frac{J(T)}{S(T)} \mathcal{D}_T - \cdots \left(\overline{\frac{\partial \varphi_{h_2}}{\partial y}}\right)_T\right] \qquad (4.3\text{-}2)$$

$$f(x) = f_0(x) + \Delta f(x)$$

In order to start the above iteration process, we need to know not only an initial guess for f, but also for β' and δ'. A discussion of the "physical" reprsentations of β and δ is given in Appendix D.

4.4 Numerical method

A computer program is written to solve the inverse problem using "partial smoothing" (refer to section 3.4). As mentioned in section 4.3, the iteration process for f consists of, in "partial smoothing" forms:

1. Guess β', δ', f as:

$$\beta' = \bar{V}'_{TX} \left(\sin 2\pi X - 1 \right)$$

$$\delta' = 0$$

$$f = f_0(x) + \frac{2}{b}\left[J(T) + S(T) I(T) \right] \left(G'_T + \delta' - A_T f' \right)$$

2. Compute

$$A_T(x) = \frac{f'(G'_T + \delta') + (\bar{V}'_{TX} + \beta')}{(1 + f'^2)}$$

$$B_T(x) = \frac{-(G''_T + \delta'') + A_T f'' + 2A'_T f'}{(1 + f'^2)}$$

3. Update

$$f(x) = -\frac{2}{b}\left[J(T) + S(T) I(T) \right] \left(G'_T + \delta' - A_T f' \right)$$

$$- \frac{2}{b} (FSUM)$$

4. Check for convergence. If not, go back to step 2 to recompute A_T, B_T with the updated values of β', δ' and f and continue this process until convergence in f is achieved.

As in the "Zero-Thickness" problem, the convergence criteria for f used during the iteration process is, at every location X_i, we require

$$f^{n+1}(x_i) - f^{n}(x_i) \leq \text{ERROR}$$

for convergence to succeed.

After convergence has been achieved, we proceed to calculate the pressure coefficients on the blade surfaces defined by:

$$C_p = 1 - \left(\frac{V}{V_{onset}} \right)^2$$

where V is defined in equation (4.2-16), and V_{onset} is defined as

$$V_{onset} \equiv \sqrt{1 + \frac{1}{4} (\tan\alpha_1 + \tan\alpha_2)^2}$$

Finally, in order to accelerate convergence, we pre-calculate all leading and trailing edges variables needed in the iteration process. The flow chart for the computer program is shown in table 4.

The above iteration process requires computing derivatives. Two methods of computing derivatives were investigated: the Spectral method (Chebyshev collocation) [7], [8], and the finite difference method (central difference). The finite difference method was chosen over the Spectral method method because it is numerically more stable.

Numerical problems were encountered in the iteration process for f. Due to "partial smoothing", this method is not able to resolve the singular point in the source/sink distribution at the leading edge. It was found that when approximately 11 points (depending on the value of the spacing δ) are used in the calculation, convergence in f is achieved in about 10 iterations. When more than 11 points are used in the calculation, the iteration process fails to converge. We propose to use a filter to resolve this problem. Studies of this numerical problem and the filtering method are discussed in Appendix E.

4.5 Design choices

In our method, there are two main input parameters available to the designer: The blockage distribution $T(x)$ and the loading distribution $\bar{V}'_{Ty}(x)$. As an example, we use analytical forms for both $T(x)$ and $\bar{V}'_{Ty}(x)$ as inputs to our numerical code.

4.5.1 Blockage distribution

In our numerical example, the blockage distribution $T(x)$ is chosen to be of the form

$$T(x) \sim x^{a}(1-x)^{b}$$

We define the maximum blockage parameter BLOCK as

$$\text{BLOCK} \equiv \frac{2\,T_{max}}{\delta}$$

We restrict ourselves to the case where $T' = 0$ at the trailing edge. If $T'(1) \neq 0$, then a stagnation point must exist at the trailing edge. We do not think that this is a good model of the real flow. In the actual low speed flow situation, the potential flow outside the boundary layer is pushed away from the trailing edge by the presence of the wake. Therefore, we think that the condition $T'(1) = 0$ is a better model for the real flow giving more realistic pressure coefficients at the trailing edge. This condition require $a > 1$. We note that this is not the restriction of our method. b can be any real number.

4.5.2 Loading distribution

In our numerical example, the loading distribution is taken to be of the form

$$\bar{V}_{Ty}' \sim \Delta p \sim x^c (1-x)^d$$

where c and d can be any real numbers.

With these design choices, we will be able to study some effects of the blockage and loading distributions on the pressure coefficients at the blade surfaces. A more practical way would be to read in the thickness and loading distributions at discrete points and use a numerical method to compute their derivatives and integrals.

4.6 Numerical results

In this section, we will first discuss the limitations of our current numerical code. Then, we will attempt to close the loop for our method using a direct method. Finally, some results are presented.

Limitations of the current numerical code

Our current numerical code (using the iteration scheme of section 4.4) is limited to the following cases:

1. when the axial chord is divided into 10 intervals or less ($<$ 11 points), f converges without using the filter. Otherwise, the filter is needed for convergence in f to succeed.

2. when the spacing to chord ratio s is of the order of 1 and greater, f converges slow and fails to converge when the convergence criteria ERROR is less than 10^{-4}.

These problems can be resolved if many more terms in the smoothing series (equation(4.2-3)) are kept so that the singular point at the leading edge can be resolved. However, we decide not to do so because the idea of the "smoothing" technique is to be able to achieve high accuracy using very few terms in the smoothing series.

Closing the loop

Our design method is supposed to find the blade shape which is supposed to do two specified jobs: a certain amount of circulation, and a certain loading distribution. Given these two parameters and the blockage distribution, our numerical code computes the corresponding blade shape and pressure coefficients at the blade surfaces.

Does this blade profile actually do the specified jobs? In an attempt to answer this question, we use a direct method to compute the circulation and the pressure coefficients of the blade shape obtained from our indirect method. If there are agreements in these results between the two methods, then we succeed to close the loop for our design method.

We choose the direct method developed by McFarland [10]. It makes use of the panel method. The numerical code has many options, some of which are what we need to perform the comparason.

In general, results of the comparason show that:

- for the individual pressure coefficients, there are good agreements between the two methods ($\sim$ 5%) away from the leading and

trailing edges ($.1 < X < .9$). Near the edges, the two results differ substantially (see example discussed below).

- for the loading (or pressure difference across the blade) distribution, the two results agree within 5%.

- for the circulation (or comparing the oulet angle), the two results agree well within 5%.

An example of closing the loop for our design problem is now presented We have taken the case where:

- inlet angle $\alpha_1 = 0^\circ$
- outlet angle $\alpha_2 = 45^\circ$
- BLOCK = .1
- spacing $s = .5$
- $T \sim X(1-X)^2$
- $\Delta p \sim X^{0.5}(1-X)$

Table 4 and 5 show the numerical results of our indirect method using 11 points and 41 points respectively. Note that the two results agree well within the error of the numerical scheme used to compute derivatives. Figures 3 and 4 show the corresponding blade shapes and pressure coefficients.

The pressure coefficients for the above blade shape (obtained from our indirect method) are calculated using the direct method. Two options in the direct method program are used:

1. Option 1 - the inlet flow condition is specified, and the program uses the Kutta condition to determine the outlet flow condition.

2. Option 2 - both the inlet and outlet flow conditions are specified (or the circulation is specified).

Table 6 shows the results obtained using option 1. The results show that:

- the exit flow angle is 45.42°, compared to the specified outlet angle of 45°.
- the loading distribution is of the specified shape (figure 5).

Table 7 shows the results obtained using option 2. The results are very similar to those obtained using option 1.

Figure 6 shows a comparason of the pressure coefficients obtained from the direct and indirect methods. It shows that the two results agree well away from the leading and trailing edges. Near these edges, the pressure coefficients obtained from the direct method show oscillations. At the trailing edge, because our blade shape is thin, the direct method fails to fit a curve through the control points giving negative thickness. Consequently, the Kutta condition at the trailing edge is not satisfied, and the results near the edges are not to be trusted.

We conclude that the blade shape obtained from our design method succeed to do the two specified jobs. We were unable to close the loop completely, but the individual pressure coefficients obtained from the two methods do have the same general shape.

Comparason of the "Zero-Thickness" and "Finite-Thickness" results in the zero blockage limit

It is found that, in the limit of $T(x)$ going to zero, the blade camber lines obtained from the two theories are not the same. Although the two theories themselves are the same in this limit, the results are not the same

because the "Zero-Thickness" problem does not predict the presence of a stagnation point at the leading edge while the "Finite-Thickness" problem does.

Table 8 shows an example of how the camber lines obtained from the two theories compare. Plots of these two blade camber lines are shown in figure 7.

Effects of the spacing to chord ratio on the blade camber

Figure 8 shows the effects of the spacing to chord ratio δ on the blade camber line. As δ increases, a smaller number of blades are available to do the same job resulting in highly cambered blades.

Effects of the maximum blockage on the pressure coefficients

Figure 9,10,11 show the effects of increasing the parameter BLOCK (the maximum blockage) on the pressure coefficients. We have taken the case where:

- inlet angle = 0°
- outlet angle = 45°
- spacing to chord $\delta = 0.5$
- $T \sim X(1-X)^2$
- $\Delta p \sim X(1-X)$

As the parameter BLOCK increases, the flow near the maximum thickness location (in this example $X^* = .33333$) accelerates due to the Venturi effect. This effect can result in highly unfavorable pressure gradients at both the upper and lower surfaces if the loading distribution is not chosen

properly. Figure 11 shows such a situation. A better loading distributionshould be sought for this example to give more acceptable pressure coefficients.

Rounded leading edge example

Figure 12 shows an example of an inlet guide vane having a rounded leading edge. Compare with figure 10 (this is a fair comparason since all inputs in the two cases are the same except for the blockage distribution), we see that the effect of the rounded leading edge is to further overexpand the fluid near the leading edge. However, we do not think the current numerical code can resolve the rounded leading edge case accurately because of the limitations on the size of the computational intervals.

4.7 Design procedure

The numerical code outlined in section 4.5 can be used to explore the effects of blade loading and blockage distributions on the behavior of the pressure coefficients at the blade surfaces. Given some design requirements, the designer can use the above program to find the "best" blade shape using a trial and error method.

In this section, we present an example of a design procedure using our numerical code. Suppose that we wish to design inlet guide vanes which can do the following jobs:

- the flow is to be turned from $\alpha_1 = 0^\circ$ to $\alpha_2 = 45^\circ$, with a spacing to chord ratio $\Delta = .5$.

- the blockage distribution is to have an analytical shape of the form $X\ (\ 1\ -\ X\ )^2$ with the maximum blockage parameter BLOCK = 0.1.

We seek a loading distribution which gives "good" pressure coefficients at the blade surfaces (in terms of minimizing flow separation). We start the trial and error process by choosing the loading distributions which is

1. highly loaded near the leading edge (maximum at $X_* = .33333$). Figure 13 shows the corresponding blade shape and pressure coefficients.

2. highly loaded at midchord. Figure 14 shows the corresponding blade shape and pressure coefficients.

3. highly loaded near the trailing edge (maximum at $X_* = .66666$). Figure 15 shows the corresponding blade shape and pressure coefficients.

Comparing the results, we conclude that, for the above example, case 1 gives the "best" pressure coefficients. In general, results show that by loading high near the leading edge, the corresponding pressure coefficients are "good". Figure 16 shows an example of a compressor blade after going through the above trial and error process. Figure 17 shows an example of an impulse blade going after going through the same process.

Chapter 5 : Conclusion

A two-dimensional design method for highly-loaded blades was presented in this thesis. Singularities are distributed on the blade camber lines to model the presence of the blades. The non-linear theory is based in part on the Clebsh formulation. A "smoothing" technique was used to solve for the blade boundary conditions. Numerical examples was presented using a "partial smoothing" form of the iteration scheme for the blade camber lines.

It was found that when the blades are assumed to be infinitely thin, the blade camber lines can be solved through an iteration process of a set of algebriac equations. The iteration process converges very fast ($\sim$7 seconds CPU time on the Digital VAX-11 computer) for the typical solidity range found in turbomachines. The results compare very well with those obtained from an "exact" method.

When the blades are assumed to have finite thickness, the "partial smoothing" form of the iteration scheme for the blade camber lines fails to resolve the singular point at the blade leading edge accurately. In order to get high accuracy, an infinite number of terms in the smoothing series would have to be kept, making the "smoothing" technique less attractive compare to other techniques. A practical numerical code based on a "partial smoothing" form of the iteration scheme for the blade camber lines was presented giving very fast convergence solutions ($\sim$10 seconds CPU time on the Digital VAX-11 computer) with satisfactory accuracy.

References

1. Lewis, R.I., "A Method for Inverse Design Aerofoil and Cascade Design by Surface Vorticity," ASME Paper No. 82-GT-154.

2. Martensen, E., "The Calculation of the Pressure Distribution on a Cascade of Thick Aerofoils by means of Fredholm Integral Equations of the Second Kind," NASA TT F-702.

3. Kashiwabara, Y., "Theory on Blades of Axial, Mixed, and Radial Turbomachines by Inverse Method," Bulletin of the JSME, February 1973.

4. Tan, C.S., Hawthorne, W.R., McCune, J.E., and Wang, C., "Theory of Blade Design for Large Deflections. Part 1: Two-Dimensional Cascade , Part 2: Annular Cascades," submitted for presentation at the 1983 ASME Gas Turbine Conference in Pheonix, Arizona.

5. McCune, J.E., "A 3D Theory of Axial Compressor Blade Rows - Subsonic, Supersonic and Transonic," Ph.D. Thesis Cornell University (1958) [see also USAFOSR TN-58-72, J. Aerospace Sc., 544 and 616 (1958)].

6. Okuroumu, O. and McCune, J.E., "Lifting Surface Theory of Axial Compressor Blade Rows: Part 1 - Supersonic Compressor, Part 2 - Compressor," AIAA J.12. 1363 and 1372 (1974).

7. Metcalfe, R.W.,"Spectral Methods for Boundary Value Problems in Fluid Mechanics," Ph.D. Thesis, Department of Mathematics, MIT, September 1973.

8. Tan, C.S., Private Communication

9. Subroutine LINV3F, chapter 2, "IMSL Library Reference Manual," Edition 9, IMSL, Inc., June 1982.

10. McFarland, E.R., "Solution of Plane Cascade Flow Using Improved Surface Singularity Methods," NASA Technical Memorandum 81589, March 1981.

11. Hornbeck, R.W., "Numerical Methods," Quantum Publishers, Inc., New York 1975.

```
INPUT

SPACING S          =   0 75
INLET ANGLE        =   0 000
OUTLET ANGLE       =  45 000
NUMBER OF POINTS   =  21
PARABOLIC LOADING INPUT PROPORTIONAL TO  x(1-x)


CONVERGENCE HISTORY OF F(X)

     ITER # 1 ----- ERRMAX = 0 01456   AT X =  0 90000
     ITER # 2 ----- ERRMAX = 0 00336   AT X =  0 60000
     ITER # 3 ----- ERRMAX = 0 00127   AT X =  0 45000
     ITER # 4 ----- ERRMAX = 0.00039   AT X =  0.55000
     ITER # 5 ----- ERRMAX = 0.00027   AT X =  0 45000
     ITER # 6 ----- ERRMAX = 0 00013   AT X =  0 40000
     ITER # 7 ----- ERRMAX = 0.00011   AT X =  0 45000
     ITER # 8 ----- ERRMAX = 0.00007   AT X =  0 50000
     ITER # 9 ----- ERRMAX = 0.00006   AT X =  0 45000
     ITER #10 ----- ERRMAX = 0.00004   AT X =  0.50000
     ITER #11 ----- ERRMAX = 0 00003   AT X =  0 45000
     ITER #12 ----- ERRMAX = 0.00003   AT X =  0.50000
     ITER #13 ----- ERRMAX = 0.00002   AT X =  0.45000
     ITER #14 ----- ERRMAX = 0 00002   AT X =  0.50000
     ITER #15 ----- ERRMAX = 0.00001   AT X =  0.45000
     ITER #16 ----- ERRMAX = 0.00001   AT X =  0.50000
     ITER #17 ----- ERRMAX = 0.00001   AT X =  0.55000
```

X	FM(X)	FLOW ANGLE	F(X)	BLADE ANGLE
0 00000	0 00000	0.00000	0.00000	-9 10520
0 05000	0 00012	0.41538	-0 00809	-9.27549
0 10000	0 00095	1.60386	-0 01633	-8.61382
0 15000	0 00312	3 47644	-0 02324	-6 47547
0 20000	0 00720	5 93741	-0.02758	-2.95814
0 25000	0 01367	8.88065	-0.02841	2.22304
0 30000	0 02295	12.18862	-0.02380	8.79703
0 35000	0 03537	15.73517	-0.01293	15.43656
0 40000	0 05120	19 39206	0.00382	21.19087
0 45000	0 07062	23.03761	0 02583	25.04913
0 50000	0 09375	25 56504	0 05269	30 19280
0 55000	0 12062	29 88813	0.08403	33 76094
0 60000	0.15120	32.94323	0 11954	36 83603
0 65000	0 18537	35.68779	0 15893	39 48203
0 70000	0 22295	38.09645	0 20192	41.75705
0 75000	0 26367	40 15600	0 24821	43.68834
0 80000	0 30720	41 86034	0 29744	45 27878
0 85000	0 35312	43.20571	0.34918	46 50883
0 90000	0 40095	44.18653	0.40285	47 31881
0 95000	0 45012	44.79155	0.45763	47 54620
1 00000	0 50000	45 00000	0 51216	47 42213

Table 1 Numerical example of 'Zero-Thickness' problem

INPUT

SPACING S = 0.75
INLET ANGLE = 0.000
OUTLET ANGLE = 45.000
PARABOLIC LOADING INPUT PROPORTIONAL TO x(1-x)

X	EXACT F	SMOOTH F	EXACT BLADE ANGLE	SMOOTH BLADE ANGLE
0.00000	0.00000	0.00000	-6.36362	-9.10520
0.05000	-0.00733	-0.00809	-8.77228	-9.27549
0.10000	-0.01498	-0.01633	-7.83866	-8.61382
0.15000	-0.02102	-0.02324	-5.31557	-6.47547
0.20000	-0.02428	-0.02768	-1.64080	-2.95814
0.25000	-0.02390	-0.02841	2.90811	2.22304
0.30000	-0.01923	-0.02380	8.06026	8.79703
0.35000	-0.00979	-0.01293	13.51490	15.43656
0.40000	0.00475	0.00382	18.96298	21.19087
0.45000	0.02450	0.02583	24.13298	26.04913
0.50000	0.04947	0.05269	28.83133	30.19280
0.55000	0.07947	0.08403	32.95549	33.76094
0.60000	0.11423	0.11954	36.48344	36.83603
0.65000	0.15338	0.15893	39.44676	39.48203
0.70000	0.19650	0.20192	41.90393	41.75705
0.75000	0.24316	0.24821	43.91742	43.68634
0.80000	0.29288	0.29744	45.53462	45.27878
0.85000	0.34517	0.34918	46.77274	46.50883
0.90000	0.39940	0.40285	47.59450	47.31881
0.95000	0.45473	0.45763	47.84251	47.54620
1.00000	0.50954	0.51216	46.68512	47.42213

Table 2 : Comparason of 'smoothing' and 'exact' results

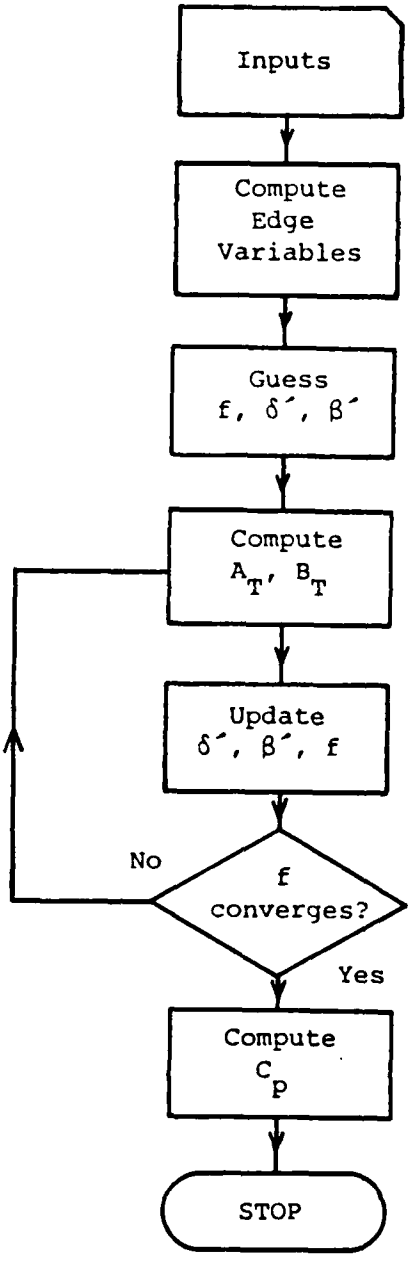


Table 3 Flow chart for the 'Finite-Thickness' problem

```
 INPUT PARAMETERS

MAX  BLOCKAGE =    0 10000
SPACING =    0 50000
INLET ANGLE =    0 00000
OUTLET ANGLE =   45 00000

NUMBER OF POINTS IJK =  11
MAX  NUMBER OF ITERATIONS ALLOWED =  20
MAX  ERROR IN F(X) ALLOWED   ERRMAX =   0.000001
FILTERING OPTION = 0

  BLOCKAGE AND LOADING PARAMETERS
    A =  1 00
    B =  2 00
    C =  0 50
    D =  1 00


          ITERATION #  1------ERRMAX =0.00376   AT X =  0 10000
          ITERATION #  2------ERRMAX =0 00048   AT X =  0.10000
          ITERATION #  3------ERRMAX =0 00013   AT X =  0.10000
          ITERATION #  4------ERRMAX =0 00002   AT X =  0 10000
          ITERATION #  5------ERRMAX =0.00000   AT X =  0 10000
          ITERATION #  6------ERRMAX =0.00000   AT X =  0.10000
```

X	LOAD(X)	T(X)	FM(X)	F(X)	Cp+	Cp-
0 00000	0 00000	0 00000	0 00000	0.00000	1 00000	1 00000
0 10000	1 09742	0 01367	0 00259	-0 01371	0 21609	-0 83669
0 20000	1 37954	0 02160	0 01335	-0 00982	0 32123	-1 00828
0 30000	1 47839	0 02481	0 03600	0 01488	0 35421	-0 91697
0 40000	1 46323	0 02430	0 07164	0 05381	0 30674	-0 86721
0 50000	1 36328	0 02109	0 12073	0 10615	0 21504	-0 85185
0 60000	1 19472	0 01620	0 18316	0 17181	0 09294	-0 83185
0 70000	0 96783	0 01063	0 25817	0 24986	-0 05235	-0 80053
0 80000	0 68977	0 00540	0 34412	0 33873	-0 21089	-0 74507
0 90000	0 36581	0 00152	0 43835	0 43595	-0 37577	-0 66037
1 00000	0 00000	0 00000	0 53713	0 53751	-0 55867	-0 55867

Table 4 Numerical example of 'Finite-Thickness' problem (without filter)

```
 INPUT PARAMETERS

MAX. BLOCKAGE =    0.10000
SPACING =    0.50000
INLET ANGLE =    0.00000
OUTLET ANGLE =   45.00000

NUMBER OF POINTS IJK =  41
MAX. NUMBER OF ITERATIONS ALLOWED =  10
MAX. ERROR IN F(X) ALLOWED   ERRMAX =   0.000100
FILTERING OPTION = 1

  BLOCKAGE AND LOADING PARAMETERS
A =  1.00
B =  2.00
C =  0.50
D =  1.00

          ITERATION #  1------ERRMAX =0.01190   AT X =  0.02500
          ITERATION #  2------ERRMAX =0.00676   AT X =  0.02500
          ITERATION #  3------ERRMAX =0.00388   AT X =  0.02500
          ITERATION #  4------ERRMAX =0.00223   AT X =  0.02500
          ITERATION #  5------ERRMAX =0.00127   AT X =  0.02500
          ITERATION #  6------ERRMAX =0.00074   AT X =  0.02500
          ITERATION #  7------ERRMAX =0.00041   AT X =  0.02500
          ITERATION #  8------ERRMAX =0.00024   AT X =  0.02500
          ITERATION #  9------ERRMAX =0.00013   AT X =  0.02500
          ITERATION # 10------ERRMAX =0.00009   AT X =  0.02500
```

X	LOAD(X)	T(X)	FM(X)	F(X)	Cp+	Cp-
0.00000	0.00000	0.00000	0.00000	0.00000	1.00000	1.00000
0.02500	0.58000	0.00401	0.00009	-0.00398	0.50002	0.24987
0.05000	0.79921	0.00761	0.00048	-0.00772	0.00085	-0.61389
0.07500	0.95307	0.01083	0.00133	-0.01066	0.03782	-0.66715
0.10000	1.07077	0.01367	0.00273	-0.01245	0.08002	-0.71882
0.12500	1.16390	0.01615	0.00474	-0.01297	0.13205	-0.75553
0.15000	1.23856	0.01829	0.00741	-0.01217	0.17935	-0.79703
0.17500	1.29845	0.02010	0.01076	-0.01010	0.22816	-0.83309
0.20000	1.34604	0.02160	0.01484	-0.00680	0.27418	-0.86304
0.22500	1.38307	0.02280	0.01966	-0.00234	0.31633	-0.88633
0.25000	1.41086	0.02373	0.02523	0.00322	0.34597	-0.90261
0.27500	1.43040	0.02439	0.03159	0.00982	0.35419	-0.91166
0.30000	1.44248	0.02481	0.03873	0.01742	0.34936	-0.91355
0.32500	1.44776	0.02499	0.04667	0.02599	0.32412	-0.90880
0.35000	1.44677	0.02495	0.05543	0.03550	0.30901	-0.89827
0.37500	1.43995	0.02472	0.06501	0.04592	0.30709	-0.88303
0.40000	1.42769	0.02430	0.07541	0.05724	0.29402	-0.86644
0.42500	1.41031	0.02371	0.08664	0.06943	0.27274	-0.85046
0.45000	1.38810	0.02297	0.09870	0.08249	0.24809	-0.84213
0.47500	1.36131	0.02209	0.11159	0.09639	0.22259	-0.83885
0.50000	1.33017	0.02109	0.12531	0.11112	0.19656	-0.83721
0.52500	1.29487	0.01999	0.13986	0.12667	0.16941	-0.83899
0.55000	1.25558	0.01879	0.15523	0.14303	0.14059	-0.83399
0.57500	1.21248	0.01753	0.17141	0.16016	0.10993	-0.82597
0.60000	1.16570	0.01620	0.18838	0.17804	0.07753	-0.82351
0.62500	1.11538	0.01483	0.20613	0.19667	0.04360	-0.81760
0.65000	1.06164	0.01344	0.22465	0.21600	0.00838	-0.81086
0.67500	1.00459	0.01203	0.24390	0.23602	-0.02793	-0.80297
0.70000	0.94433	0.01063	0.26387	0.25670	-0.06516	-0.79369
0.72500	0.88096	0.00925	0.28452	0.27800	-0.10315	-0.78285
0.75000	0.81456	0.00791	0.30582	0.29990	-0.14174	-0.77030
0.77500	0.74522	0.00662	0.32773	0.32237	-0.18080	-0.75593
0.80000	0.67302	0.00540	0.35021	0.34537	-0.22020	-0.73968
0.82500	0.59802	0.00426	0.37320	0.36888	-0.25988	-0.72153
0.85000	0.52030	0.00323	0.39667	0.39287	-0.29980	-0.70153
0.87500	0.43991	0.00231	0.42055	0.41729	-0.34003	-0.67980
0.90000	0.35692	0.00152	0.44478	0.44213	-0.38069	-0.65653
0.92500	0.27138	0.00088	0.46931	0.46732	-0.42202	-0.63202
0.95000	0.18335	0.00040	0.49406	0.49282	-0.46446	-0.60669
0.97500	0.09287	0.00010	0.51896	0.51855	-0.50899	-0.58137
1.00000	0.00000	0.00000	0.54394	0.54434	-0.55891	-0.55891

Table 5 : Numerical example of 'Finite-Thickness' problem (with filter)

CONTROL POINT	CONTROL POINT COORDINATE X	Y	SOURCE DENSITY	NORMAL VELOCITY	TANGENTIAL VELOCITY	PRESSURE COEFFICIENT
1	0.987487	0.531406	-0.327305	0.000000	-1.003151	-0.006311
2	0.962497	0.505435	-0.329590	0.000000	-1.579879	-1.496018
3	0.937506	0.479427	-0.328757	0.000000	-1.173611	-0.377364
4	0.912511	0.453515	-0.326162	0.000000	-1.317903	-0.736867
5	0.887515	0.427783	-0.322227	0.000000	-1.304845	-0.702619
6	0.862519	0.402295	-0.317136	0.000000	-1.312181	-0.721820
7	0.837522	0.377107	-0.310993	0.000000	-1.316798	-0.733958
8	0.812525	0.352270	-0.303809	0.000000	-1.321021	-0.745096
9	0.787528	0.327831	-0.295573	0.000000	-1.324733	-0.754919
10	0.762531	0.303837	-0.286240	0.000000	-1.328029	-0.763659
11	0.737534	0.280335	-0.275760	0.000000	-1.330985	-0.771520
12	0.712537	0.257369	-0.264055	0.000000	-1.333702	-0.778762
13	0.687539	0.234986	-0.251053	0.000000	-1.336256	-0.785580
14	0.662542	0.213232	-0.236684	0.000000	-1.338725	-0.792184
15	0.637544	0.192150	-0.220876	0.000000	-1.341169	-0.798733
16	0.612547	0.171783	-0.203552	0.000000	-1.343636	-0.805356
17	0.587549	0.152174	-0.184650	0.000000	-1.346141	-0.812096
18	0.562550	0.133362	-0.164110	0.000000	-1.348704	-0.819002
19	0.537552	0.115385	-0.141866	0.000000	-1.351329	-0.826091
20	0.512553	0.098279	-0.117855	0.000000	-1.354002	-0.833322
21	0.487553	0.082078	-0.092026	0.000000	-1.356702	-0.840639
22	0.462554	0.066815	-0.064315	0.000000	-1.359402	-0.847975
23	0.437553	0.052520	-0.034663	0.000000	-1.362056	-0.855198
24	0.412553	0.039225	-0.003003	0.000000	-1.364601	-0.862137
25	0.387552	0.026961	0.030739	0.000000	-1.366939	-0.868523
26	0.362550	0.015759	0.066643	0.000000	-1.368936	-0.873986
27	0.337548	0.005653	0.104813	0.000000	-1.370391	-0.877972
28	0.312545	-0.003320	0.145361	0.000000	-1.371003	-0.879651
29	0.287541	-0.011118	0.188412	0.000000	-1.370336	-0.877822
30	0.262536	-0.017693	0.234096	0.000000	-1.367744	-0.870723
31	0.237530	-0.022990	0.282509	0.000000	-1.362293	-0.855842
32	0.212523	-0.026944	0.333655	0.000000	-1.352654	-0.829674
33	0.187514	-0.029482	0.387351	0.000000	-1.337004	-0.787580
34	0.162502	-0.030524	0.443059	0.000000	-1.312949	-0.722835
35	0.137490	-0.029992	0.499690	0.000000	-1.277559	-0.632158
36	0.112476	-0.027823	0.555407	0.000000	-1.227688	-0.507219
37	0.087465	-0.023995	0.607521	0.000000	-1.159363	-0.344123
38	0.062461	-0.018570	0.652563	0.000000	-1.073582	-0.152578
39	0.037470	-0.011764	0.685564	0.000000	-0.980098	0.075357
40	0.014985	-0.004866	0.700727	0.000000	-0.993459	0.013040
41	0.002499	-0.000828	0.709200	0.000000	-0.201021	0.959591
42	0.002500	0.000011	-0.448950	0.000000	0.587520	0.654821
43	0.015000	0.000041	-0.451883	0.000000	0.925071	0.144244
44	0.037500	-0.000047	-0.457399	0.000000	0.765250	0.414393
45	0.062501	-0.000039	-0.442719	0.000000	0.771431	0.404895
46	0.087505	0.000588	-0.414927	0.000000	0.755696	0.428923
47	0.112510	0.002081	-0.381196	0.000000	0.745731	0.443885
48	0.137514	0.004531	-0.345486	0.000000	0.741692	0.449893
49	0.162518	0.007946	-0.310129	0.000000	0.742463	0.448749
50	0.187521	0.012206	-0.276276	0.000000	0.746854	0.442210
51	0.212523	0.017535	-0.244315	0.000000	0.753821	0.431754
52	0.237525	0.023614	-0.214207	0.000000	0.762583	0.418467
53	0.262526	0.030492	-0.185719	0.000000	0.772603	0.403284
54	0.287527	0.038138	-0.158562	0.000000	0.783536	0.386071
55	0.312529	0.046526	-0.132463	0.000000	0.795173	0.367700
56	0.337530	0.055641	-0.107195	0.000000	0.807401	0.348104
57	0.362531	0.065472	-0.082593	0.000000	0.820164	0.327331
58	0.387533	0.076015	-0.058547	0.000000	0.833439	0.305379
59	0.412534	0.087267	-0.034989	0.000000	0.847222	0.282214
60	0.437536	0.099228	-0.011901	0.000000	0.861516	0.257791
61	0.462537	0.111899	0.010708	0.000000	0.876309	0.232082
62	0.487538	0.125291	0.032818	0.000000	0.891593	0.205062
63	0.512539	0.139374	0.054374	0.000000	0.907342	0.176731
64	0.537540	0.154177	0.075340	0.000000	0.923510	0.147129
65	0.562540	0.169688	0.095655	0.000000	0.940051	0.116304
66	0.587541	0.185903	0.115273	0.000000	0.956890	0.084361
67	0.612541	0.202814	0.134157	0.000000	0.973947	0.051427
68	0.637541	0.220415	0.152273	0.000000	0.991135	0.017650
69	0.662540	0.238695	0.169601	0.000000	1.008358	-0.016786
70	0.687540	0.257642	0.186136	0.000000	1.025518	-0.051686
71	0.712539	0.277243	0.201884	0.000000	1.042525	-0.086858
72	0.737539	0.297486	0.216868	0.000000	1.059300	-0.122116
73	0.762538	0.318356	0.231109	0.000000	1.075786	-0.157315
74	0.787537	0.339838	0.244641	0.000000	1.091943	-0.192339
75	0.812536	0.361918	0.257491	0.000000	1.107753	-0.227118
76	0.837535	0.384581	0.269679	0.000000	1.123289	-0.261779
77	0.862534	0.407811	0.281197	0.000000	1.138090	-0.295250
78	0.887533	0.431589	0.292010	0.000000	1.156213	-0.337059
79	0.912531	0.455891	0.302015	0.000000	1.150741	-0.324205
80	0.937527	0.480684	0.311029	0.000000	1.227682	-0.507203
81	0.962521	0.505916	0.318635	0.000000	1.189794	-0.415610
82	0.987512	0.531485	0.323538	0.000000	1.242529	-0.543878

UNIFORM ERROR (EPS) = -0.7478754E-03

CL = -0.9088286
CD = 0.2866034E-02
CIRCULATION = -0.4525191

POTENTIAL FLOW VELOCITY DIAGRAM
UPSTREAM VELOCITY = 0.89175 AT 0.00004 DEGREES
ONSET VELOCITY = 1.00000 AT 26.90546 DEGREES
DOWNSTREAM VELOCITY = 1.27056 AT 45.42360 DEGREES

*** NOTE: ALL VELOCITY QUANITIES ARE SCALED BY THE ONSET VELOCITY ***

Table 6 Results from direct method (inlet condition specified)

CONTROL POINT	CONTROL POINT COORDINATE X	Y	SOURCE DENSITY	NORMAL VELOCITY	TANGENTIAL VELOCITY	PRESSURE COEFFICIENT
1	0.987487	0.531406	-0.332912	0.000000	-1.011293	-0.022692
2	0.962497	0.505435	-0.335192	0.000000	-1.520596	-1.312211
3	0.937506	0.479427	-0.334361	0.000000	-1.180097	-0.392604
4	0.912511	0.453515	-0.331771	0.000000	-1.306018	-0.705682
5	0.887515	0.427783	-0.327845	0.000000	-1.297752	-0.684161
6	0.862519	0.402295	-0.322764	0.000000	-1.306381	-0.706631
7	0.837522	0.377107	-0.316634	0.000000	-1.312422	-0.722451
8	0.812525	0.352270	-0.309463	0.000000	-1.317007	-0.736880
9	0.787529	0.327831	-0.301243	0.000000	-1.322764	-0.749704
10	0.762531	0.303837	-0.291927	0.000000	-1.327085	-0.761153
11	0.737534	0.280335	-0.281465	0.000000	-1.330945	-0.771416
12	0.712537	0.257369	-0.269780	0.000000	-1.334444	-0.780740
13	0.687539	0.234996	-0.256799	0.000000	-1.337655	-0.789322
14	0.662542	0.213232	-0.242451	0.000000	-1.340666	-0.797387
15	0.637544	0.192150	-0.226666	0.000000	-1.343548	-0.805123
16	0.612547	0.171793	-0.209365	0.000000	-1.346364	-0.812696
17	0.587549	0.152174	-0.190485	0.000000	-1.349143	-0.820187
18	0.562550	0.133362	-0.169968	0.000000	-1.351919	-0.827685
19	0.537552	0.115385	-0.147744	0.000000	-1.354710	-0.835239
20	0.512553	0.098279	-0.123753	0.000000	-1.357511	-0.842837
21	0.487553	0.082078	-0.097940	0.000000	-1.360310	-0.850443
22	0.462554	0.066815	-0.070242	0.000000	-1.363088	-0.858048
23	0.437553	0.052520	-0.040600	0.000000	-1.365801	-0.865413
24	0.412553	0.039225	-0.008944	0.000000	-1.368391	-0.872495
25	0.387552	0.026961	0.024800	0.000000	-1.370762	-0.879930
26	0.362550	0.015759	0.060714	0.000000	-1.372792	-0.884530
27	0.337548	0.005683	0.098902	0.000000	-1.374249	-0.888560
28	0.312545	-0.003320	0.139479	0.000000	-1.374863	-0.890247
29	0.287541	-0.011118	0.182574	0.000000	-1.374184	-0.888392
30	0.262536	-0.017693	0.228315	0.000000	-1.371565	-0.881191
31	0.237530	-0.022990	0.276803	0.000000	-1.366069	-0.866144
32	0.212523	-0.026944	0.328049	0.000000	-1.356360	-0.839713
33	0.187514	-0.029482	0.381865	0.000000	-1.340608	-0.797229
34	0.162502	-0.030524	0.437723	0.000000	-1.316409	-0.732933
35	0.137490	-0.029932	0.494534	0.000000	-1.280922	-0.640505
36	0.112476	-0.027823	0.550455	0.000000	-1.230687	-0.514590
37	0.087465	-0.023995	0.602789	0.000000	-1.162009	-0.350243
38	0.062461	-0.018570	0.648048	0.000000	-1.075764	-0.157269
39	0.037470	-0.011764	0.681224	0.000000	-0.969196	0.073363
40	0.014985	-0.004866	0.696473	0.000000	-0.995337	0.009303
41	0.002499	-0.000829	0.704997	0.000000	-0.192771	0.962939
42	0.002500	0.000011	-0.443632	0.000000	0.595012	0.645973
43	0.015000	0.000041	-0.446573	0.000000	0.929794	0.135483
44	0.037500	-0.000047	-0.452106	0.000000	0.768674	0.409140
45	0.062501	-0.000039	-0.437392	0.000000	0.774482	0.400178
46	0.087505	0.000588	-0.409512	0.000000	0.758529	0.424634
47	0.112510	0.002091	-0.375695	0.000000	0.748417	0.439872
48	0.137514	0.004531	-0.339004	0.000000	0.744284	0.446042
49	0.162519	0.007946	-0.304474	0.000000	0.744995	0.444982
50	0.187521	0.012296	-0.270560	0.000000	0.749350	0.438474
51	0.212523	0.017535	-0.239548	0.000000	0.756298	0.428014
52	0.237525	0.023614	-0.208399	0.000000	0.765151	0.414696
53	0.262526	0.030492	-0.179878	0.000000	0.775071	0.399265
54	0.287527	0.038138	-0.152693	0.000000	0.786010	0.382189
55	0.312529	0.046526	-0.126571	0.000000	0.797658	0.363741
56	0.337530	0.055641	-0.101286	0.000000	0.809001	0.344060
57	0.362531	0.065472	-0.076670	0.000000	0.822685	0.323190
58	0.387533	0.076015	-0.052615	0.000000	0.835983	0.301133
59	0.412534	0.087267	-0.029051	0.000000	0.849793	0.277852
60	0.437536	0.099228	-0.005959	0.000000	0.864117	0.253303
61	0.462537	0.111899	0.016648	0.000000	0.878945	0.227456
62	0.487538	0.125281	0.038756	0.000000	0.894266	0.200299
63	0.512539	0.139374	0.060306	0.000000	0.910056	0.171798
64	0.537540	0.154177	0.081262	0.000000	0.926270	0.142024
65	0.562540	0.169688	0.101567	0.000000	0.942960	0.111014
66	0.587541	0.185903	0.121172	0.000000	0.959754	0.078872
67	0.612541	0.202814	0.140042	0.000000	0.976870	0.045724
68	0.637541	0.220415	0.158141	0.000000	0.994125	0.011716
69	0.662540	0.238695	0.175452	0.000000	1.011421	-0.022972
70	0.687540	0.257642	0.191970	0.000000	1.028664	-0.058149
71	0.712539	0.277243	0.207699	0.000000	1.045767	-0.093629
72	0.737539	0.297486	0.222664	0.000000	1.062656	-0.129239
73	0.762538	0.318356	0.236885	0.000000	1.079291	-0.164848
74	0.787537	0.339838	0.250397	0.000000	1.095612	-0.200367
75	0.812536	0.361918	0.263226	0.000000	1.111651	-0.235768
76	0.837535	0.384581	0.275394	0.000000	1.127487	-0.271227
77	0.862534	0.407811	0.286892	0.000000	1.142781	-0.305949
78	0.887533	0.431589	0.297687	0.000000	1.161149	-0.348266
79	0.912531	0.455891	0.307673	0.000000	1.159661	-0.344814
80	0.937527	0.480694	0.316670	0.000000	1.229022	-0.510496
81	0.962521	0.505916	0.324260	0.000000	1.201048	-0.442516
82	0.987512	0.531485	0.329153	0.000000	1.267424	-0.606364

UNIFORM ERROR (EPS) = -0.7129127E-03

CL = -0.8971415
CD = 0.2562509E-02
CIRCULATION = -0.4472136

POTENTIAL FLOW VELOCITY DIAGRAM
UPSTREAM VELOCITY = 0.89443 AT 0.00000 DEGREES
ONSET VELOCITY = 1.00000 AT 26.56505 DEGREES
DOWNSTREAM VELOCITY = 1.26491 AT 45.00000 DEGREES

*** NOTE: ALL VELOCITY QUANITIES ARE SCALED BY THE ONSET VELOCITY ***

Table 7 Results from direct method (circulation specified)

INPUT

BLOCK = 0.00001
SPACING S = 0.75
INLET ANGLE = 0.000
OUTLET ANGLE = 45.000
PARABOLIC LOADING INPUT PROPORTIONAL TO x(1-x)

X	F(X) finite thickness result	F(X) zero thickness result
0.00000	0.00000	0.00000
0.05000	-0.00849	-0.00809
0.10000	-0.01803	-0.01633
0.15000	-0.02680	-0.02324
0.20000	-0.03334	-0.02768
0.25000	-0.03620	-0.02841
0.30000	-0.03347	-0.02380
0.35000	-0.02376	-0.01293
0.40000	-0.00761	0.00382
0.45000	0.01410	0.02583
0.50000	0.04083	0.05269
0.55000	0.07215	0.08403
0.60000	0.10770	0.11951
0.65000	0.14715	0.15893
0.70000	0.19020	0.20192
0.75000	0.23654	0.24821
0.80000	0.28581	0.29744
0.85000	0.33757	0.34918
0.90000	0.39123	0.40285
0.95000	0.44600	0.45763
1.00000	0.50053	0.51216

Table 8 : Comparason of 'Zero-Thickness' and 'Finite-Thickness' results in the zero blockage limit

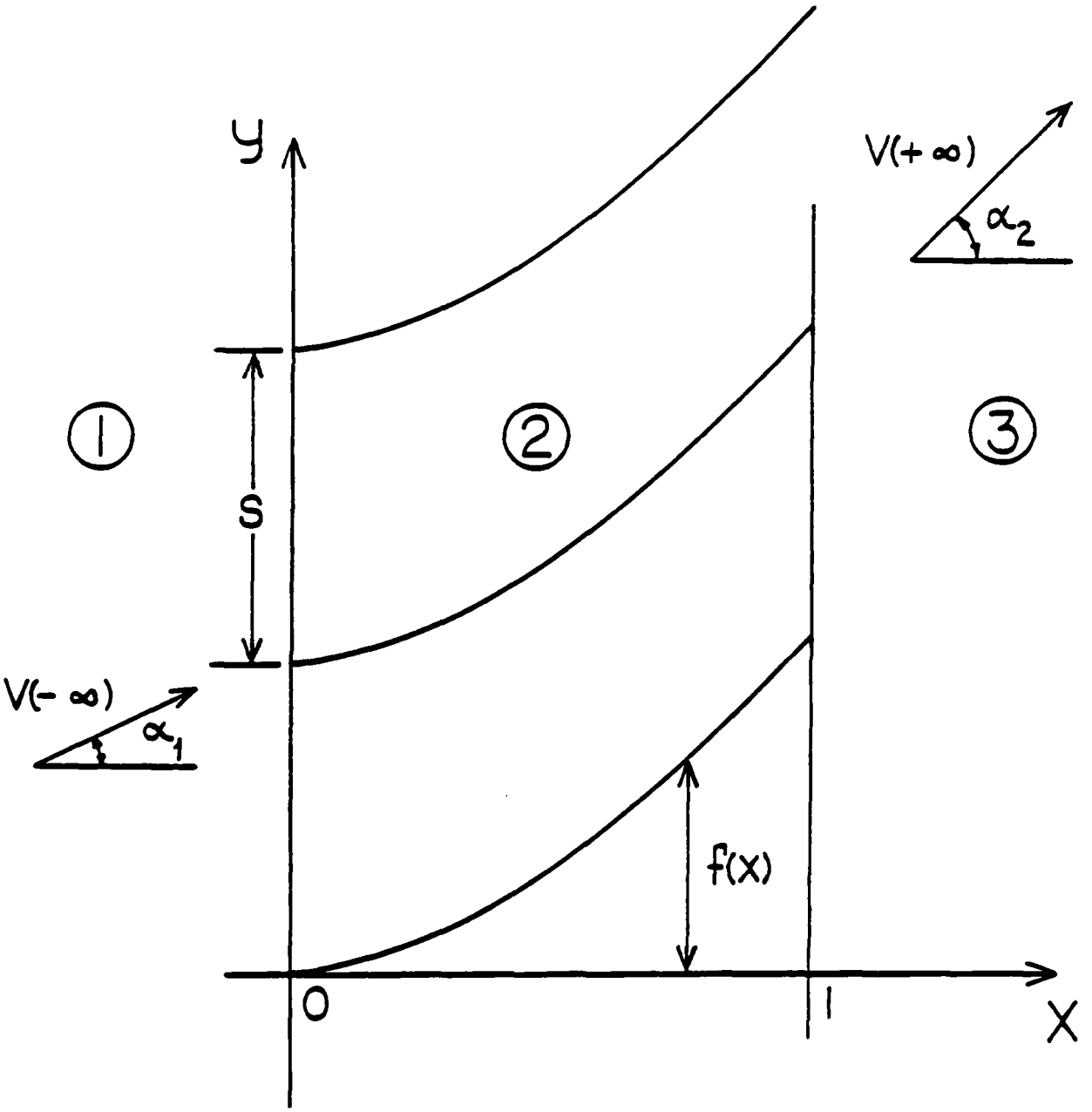


Figure 1 2D cascade notation

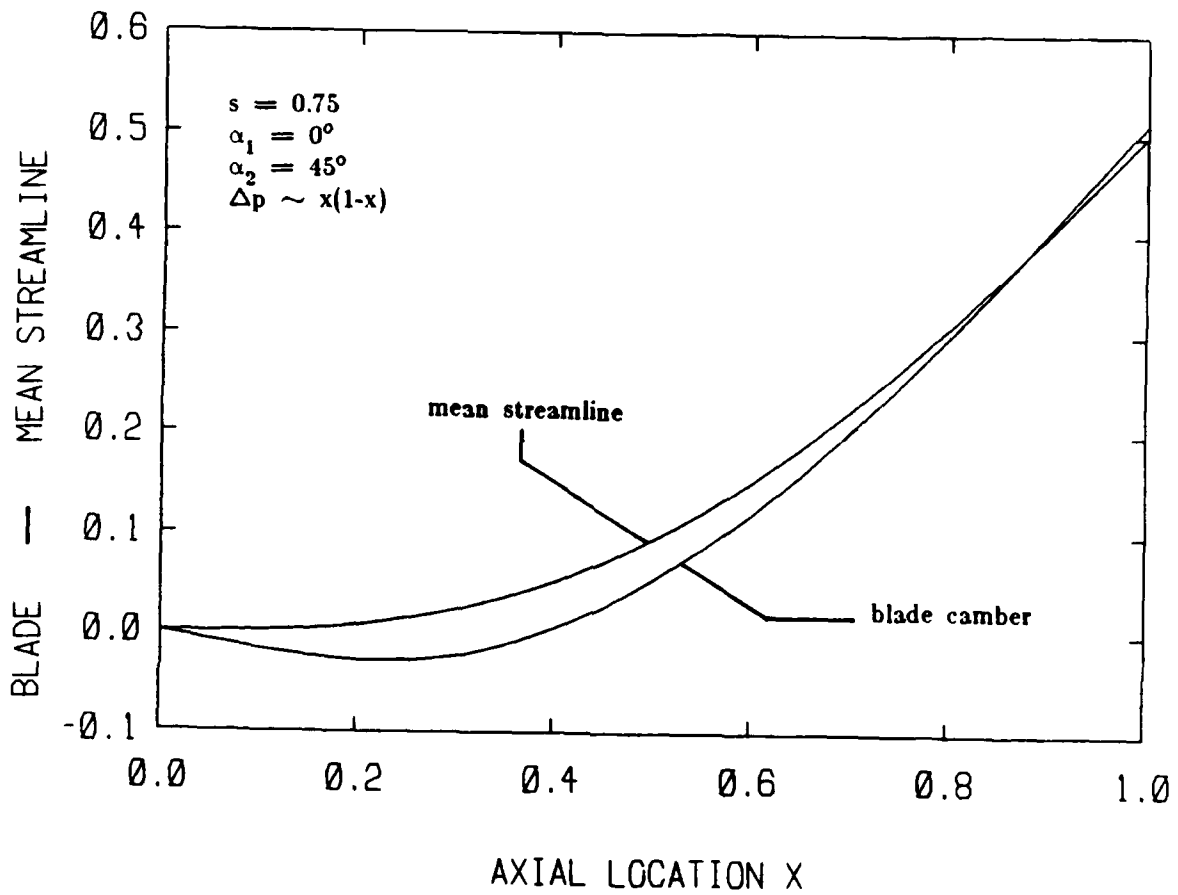


Figure 2 : Blade camber and mean streamline

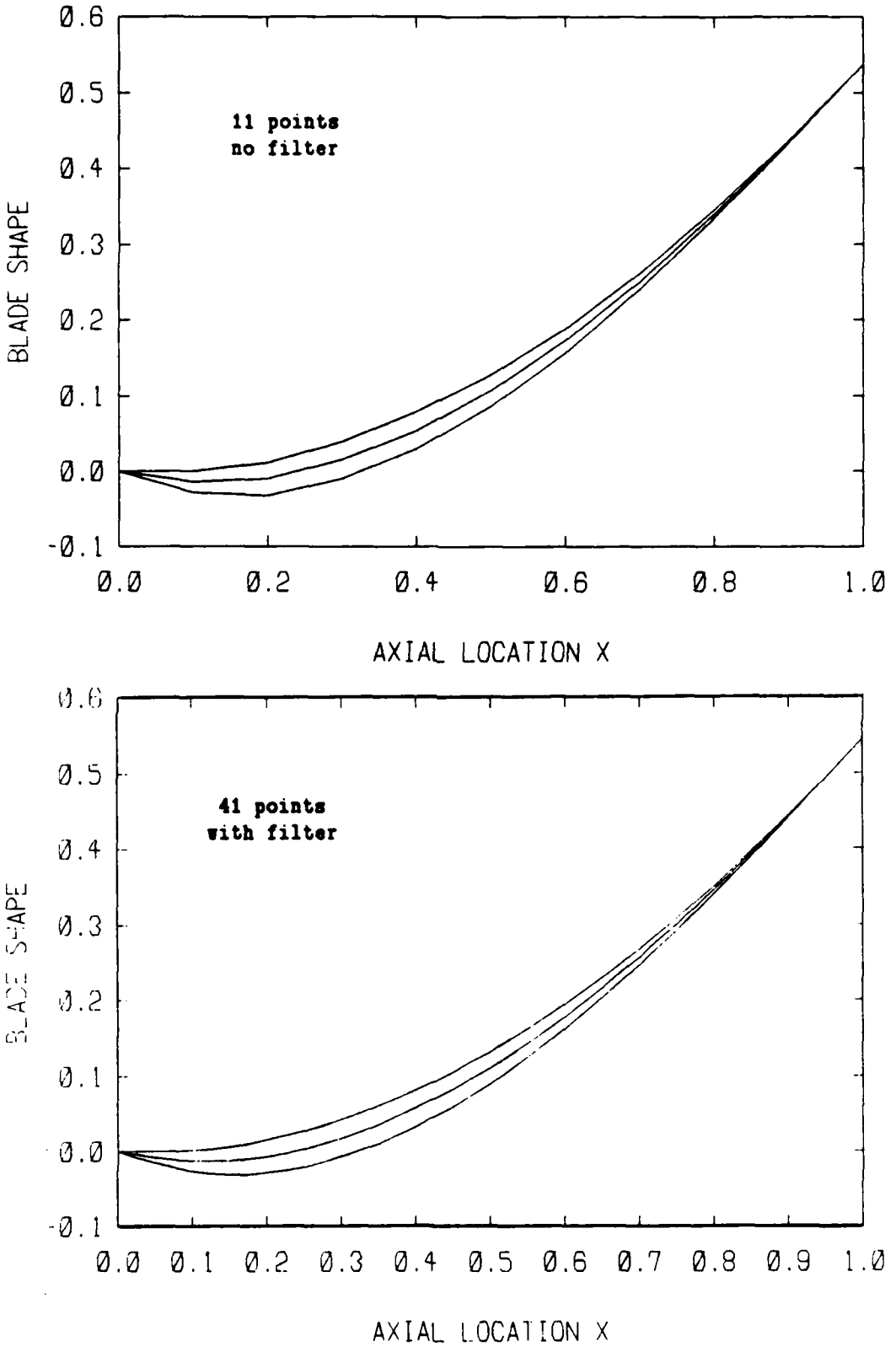
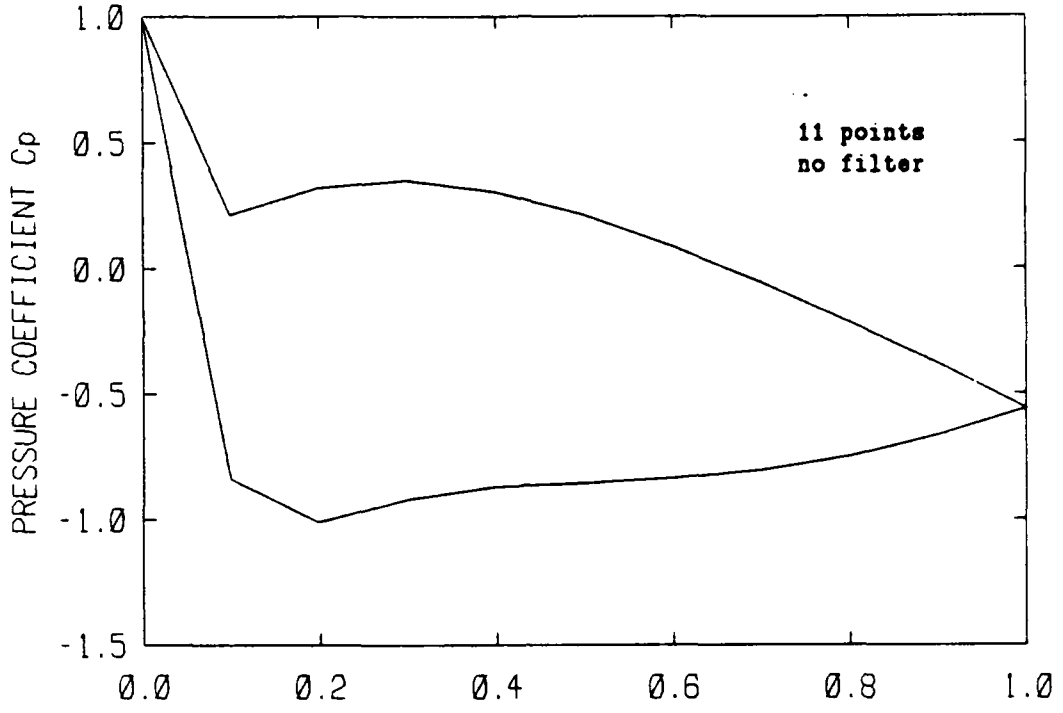


Figure 3 : Blade shapes obtained from 'Smoothing' technique



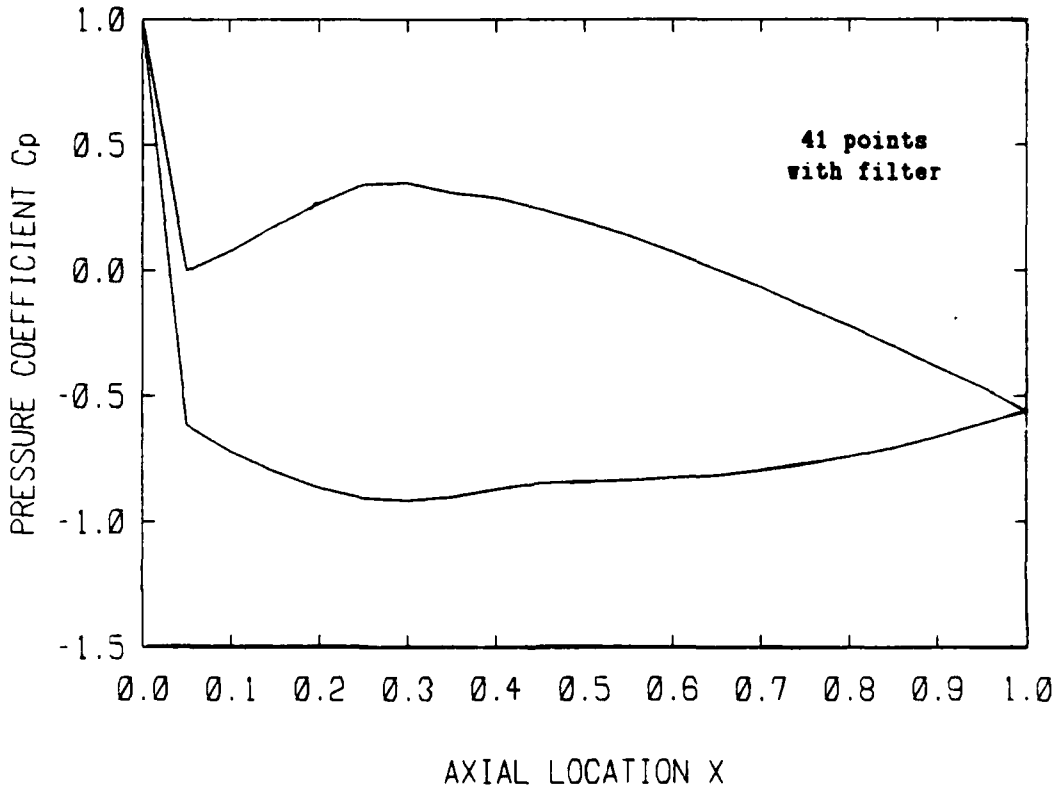
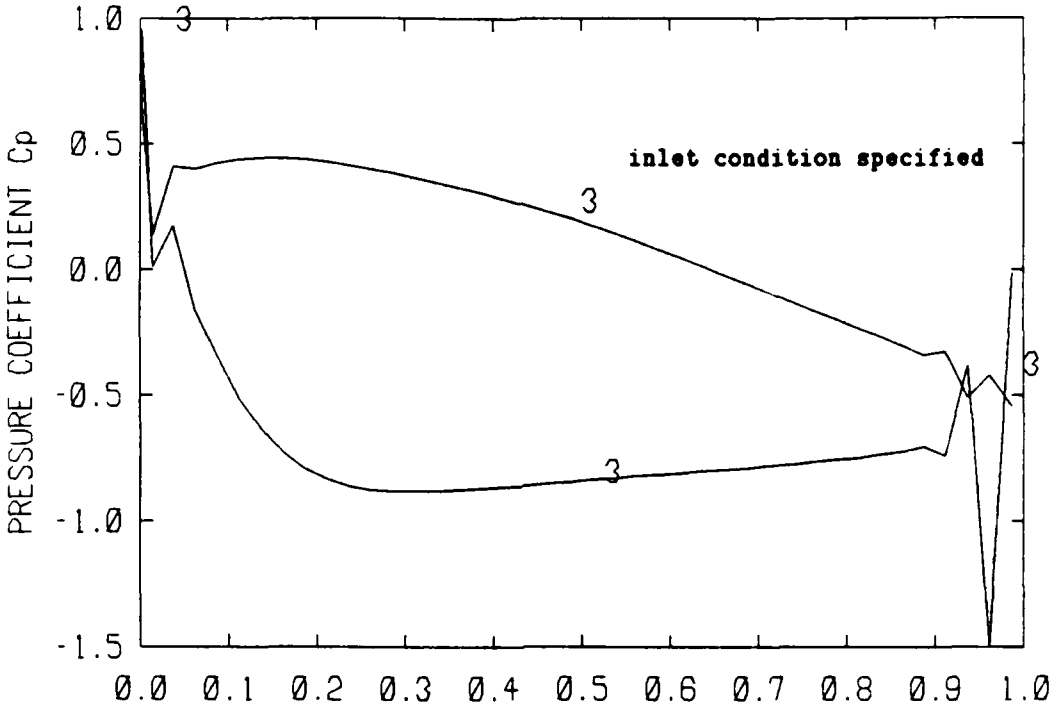


Figure 4 Pressure coefficients obtained from 'Smoothing' technique



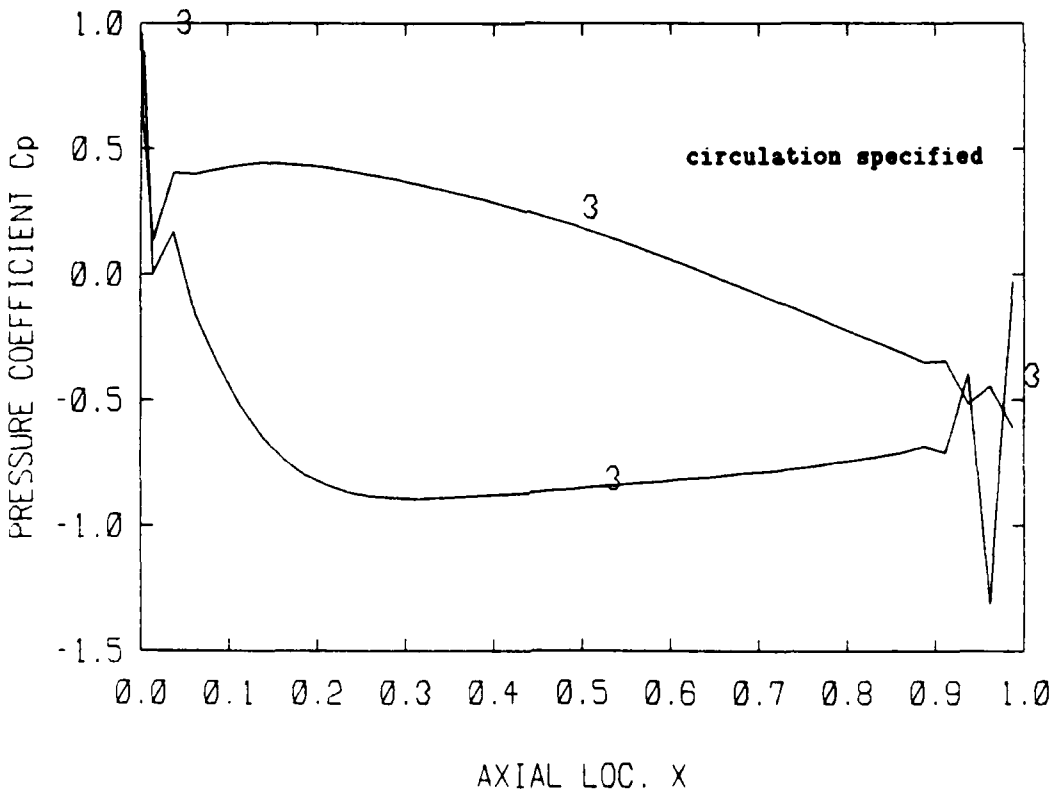
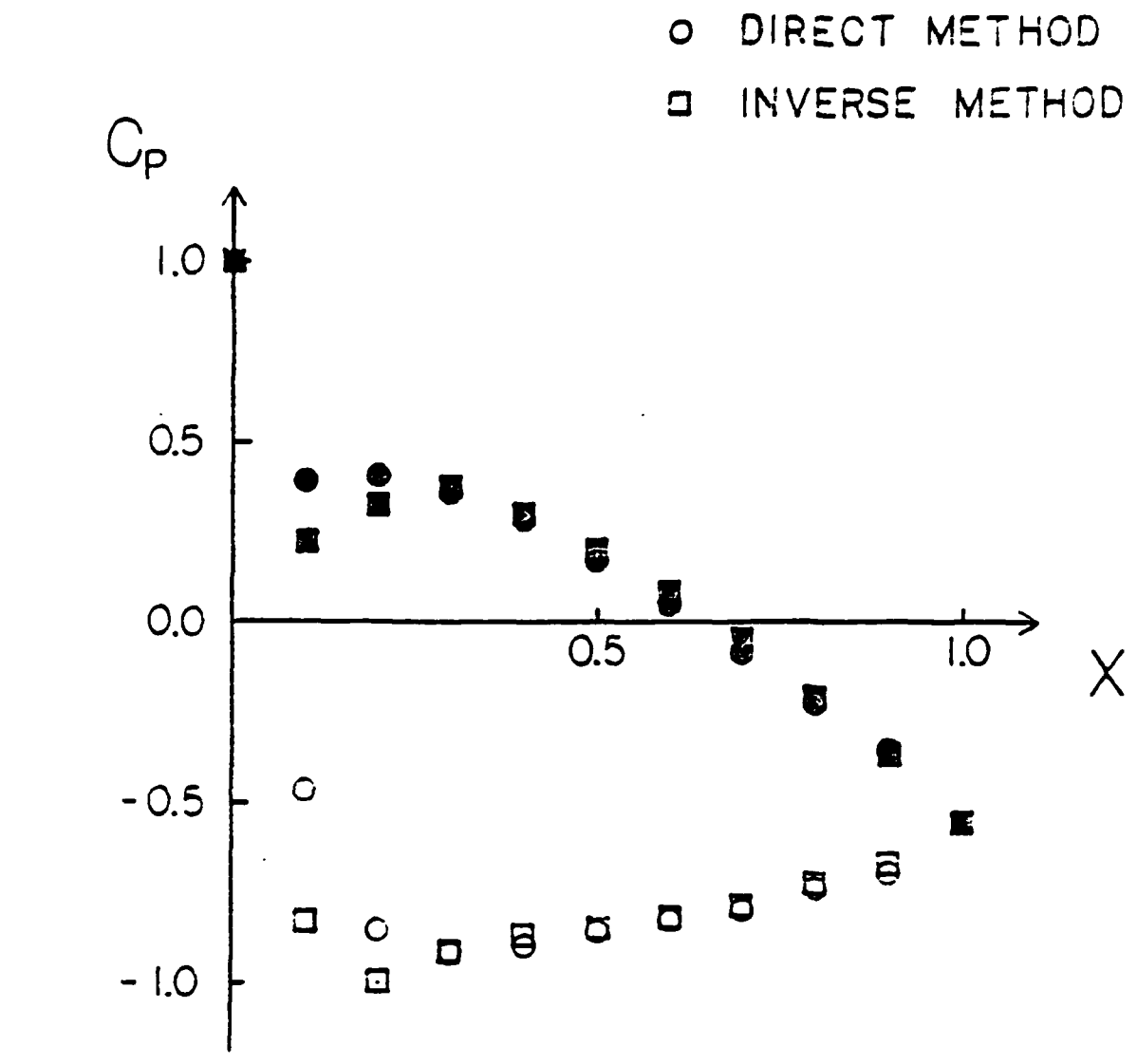


Figure 5 : Pressure coefficients obtained from direct method



$S = 0.5$

$\Delta P \sim \sqrt{X}(1-X)$

$T \sim X(1-X)^2$

BLOCK = 0.1

Figure 6 : Comparason of Cp's from direct and inverse methods

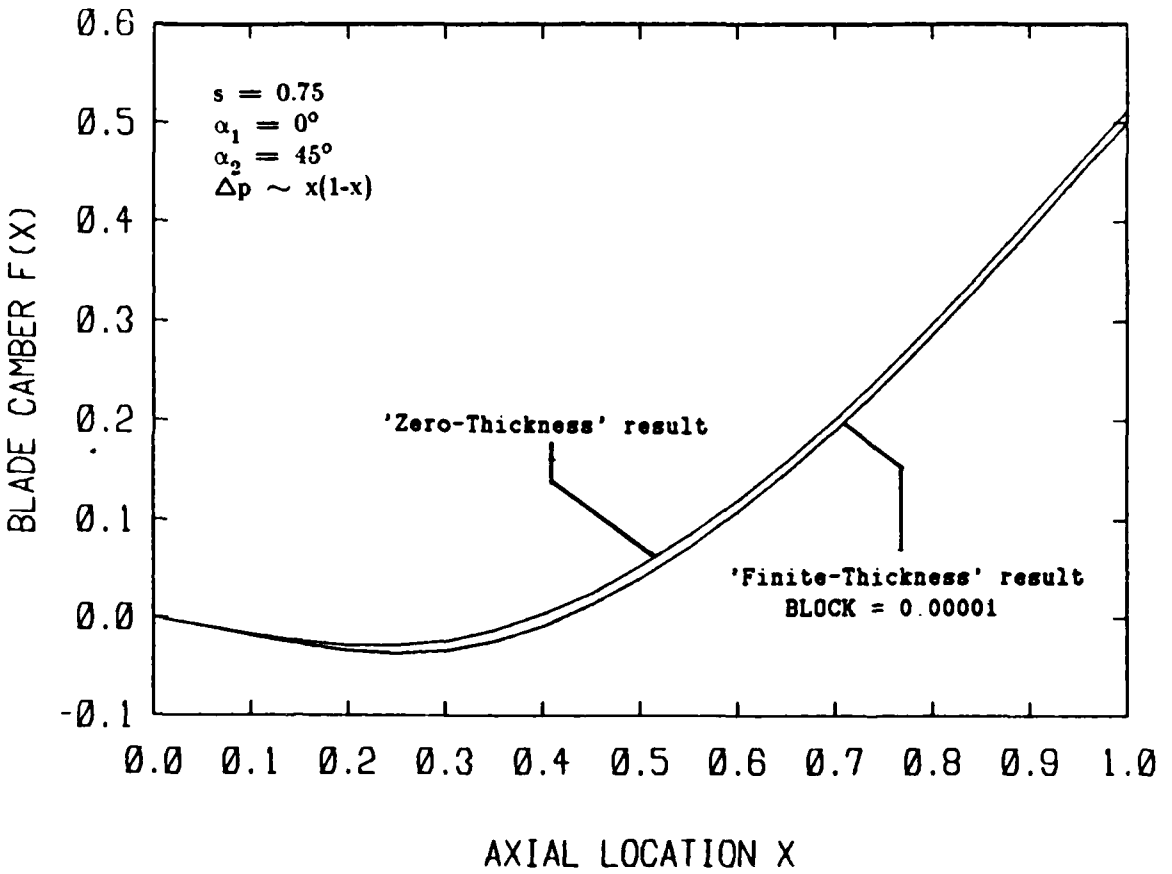


Figure 7 Comparason of 'Zero-Thickness' and 'Finite-Thickness' results in the zero blockage limit

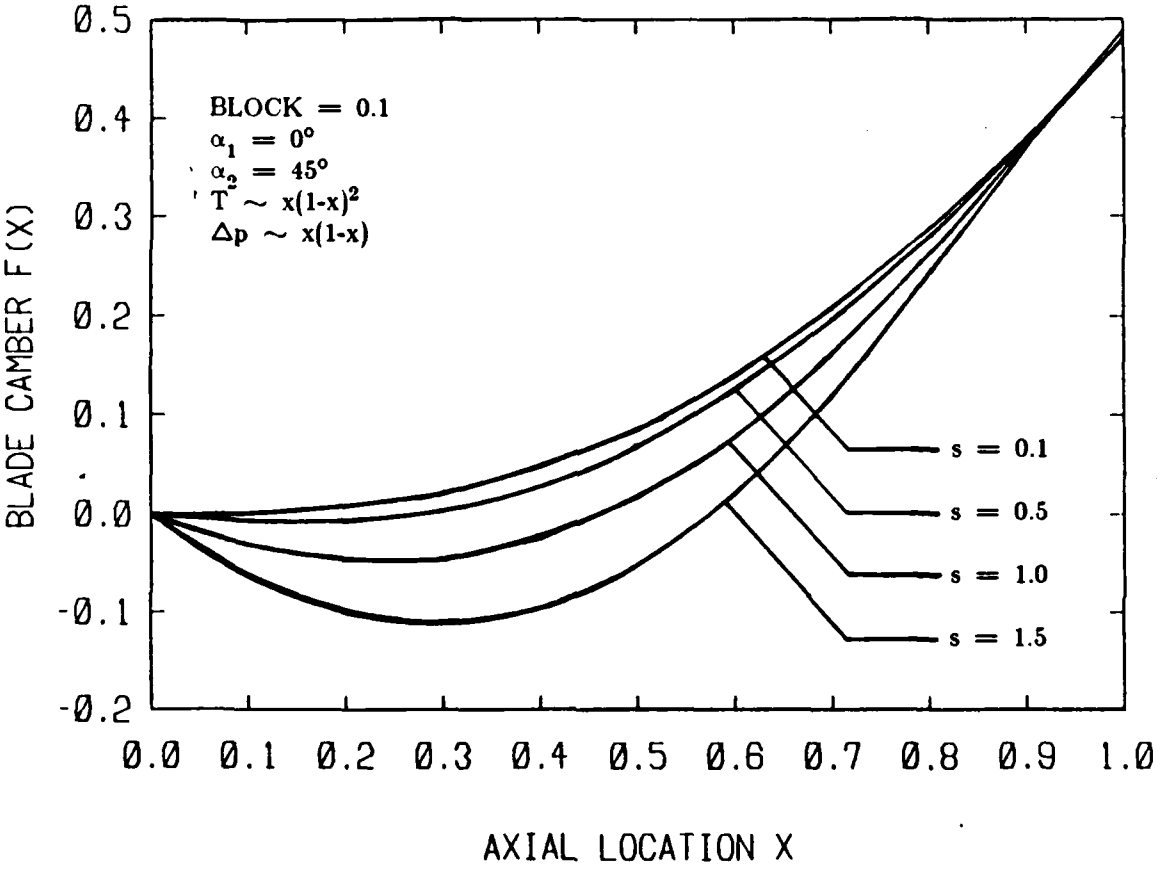


Figure 8 Effects of spacing to chord ratio on the blade cambers

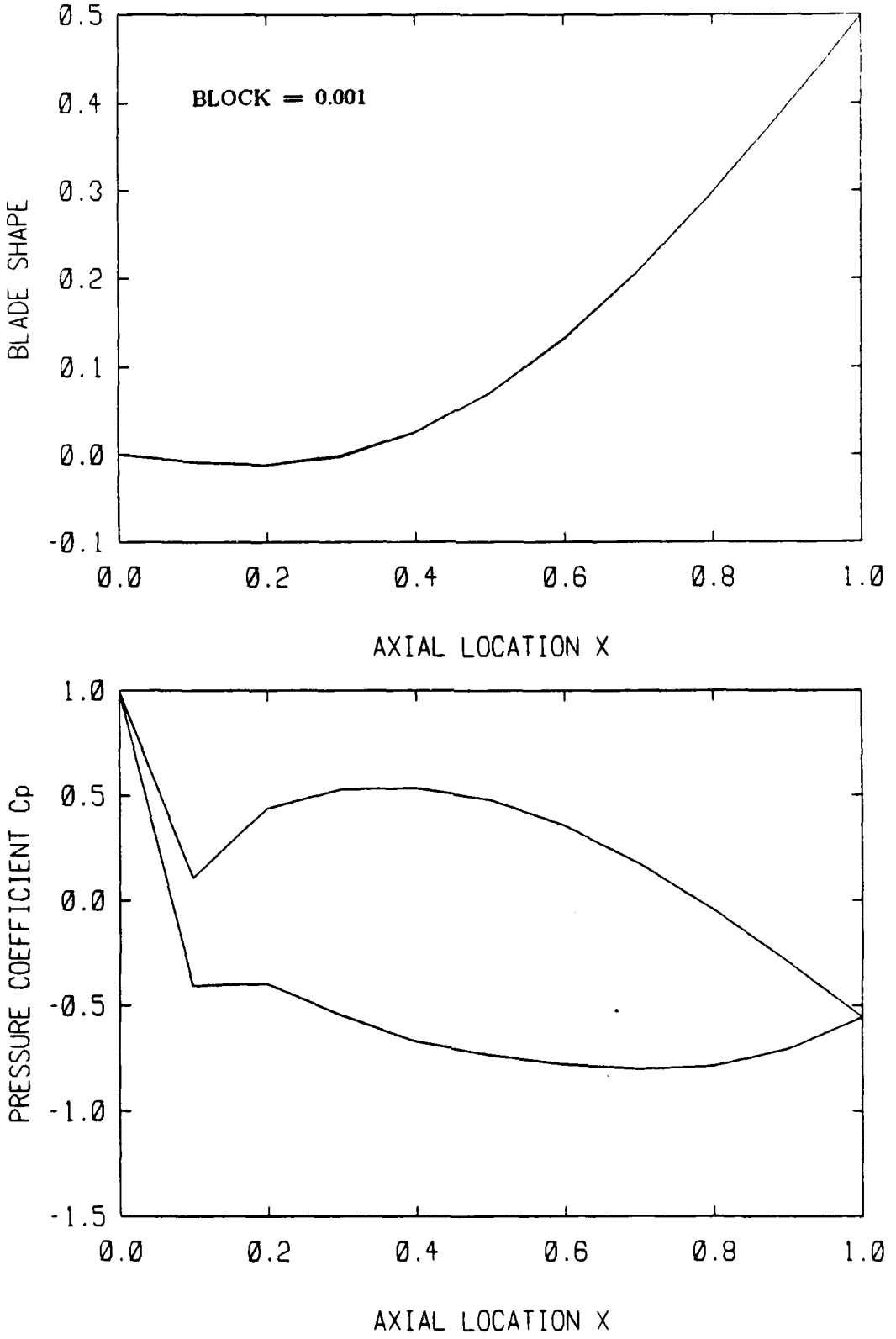


Figure 9 Effects of maximum blockage on Cp's (BLOCK = 0.001)

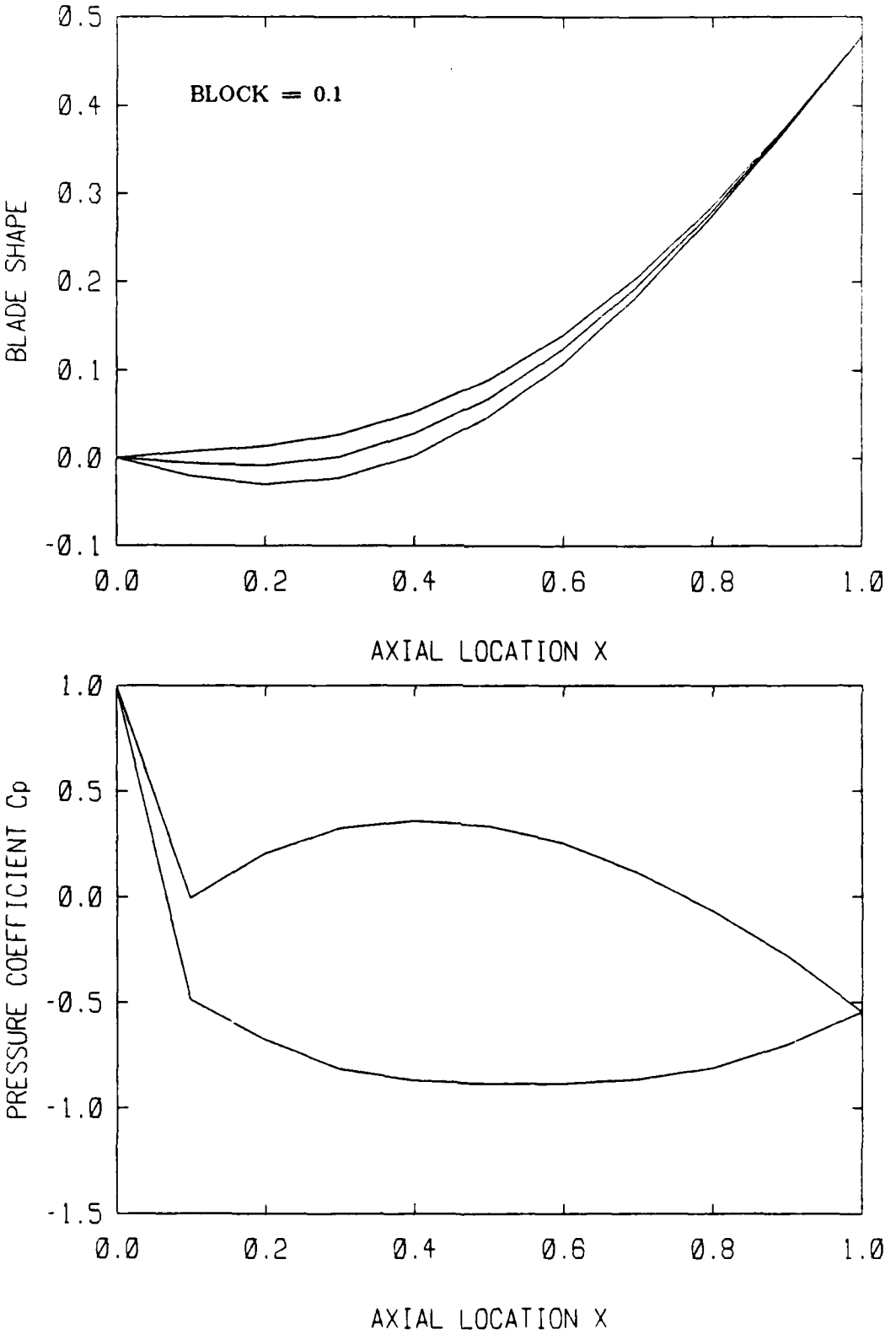


Figure 10 Effects of maximin blockage on Cp's (BLOCK = 0.1)

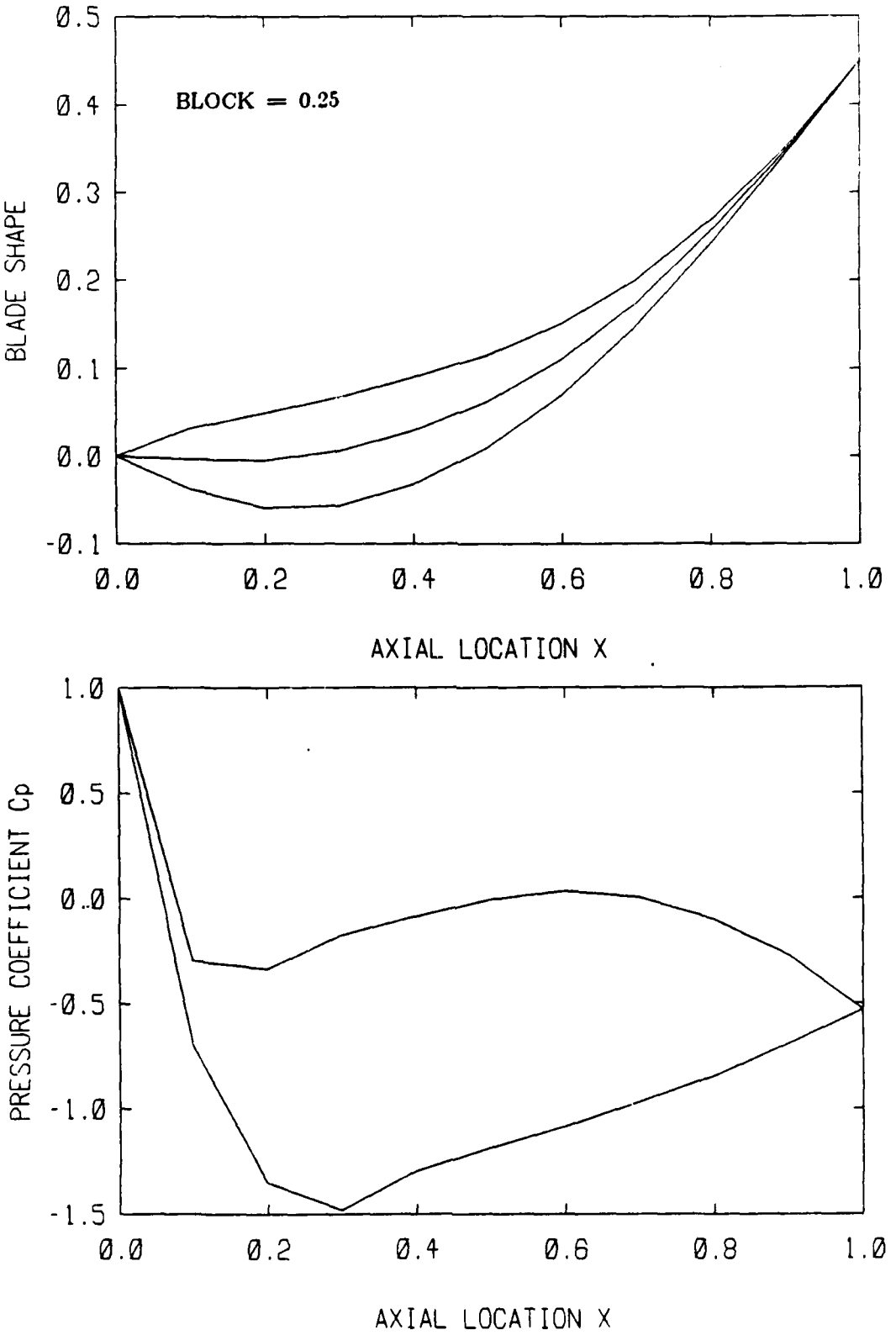
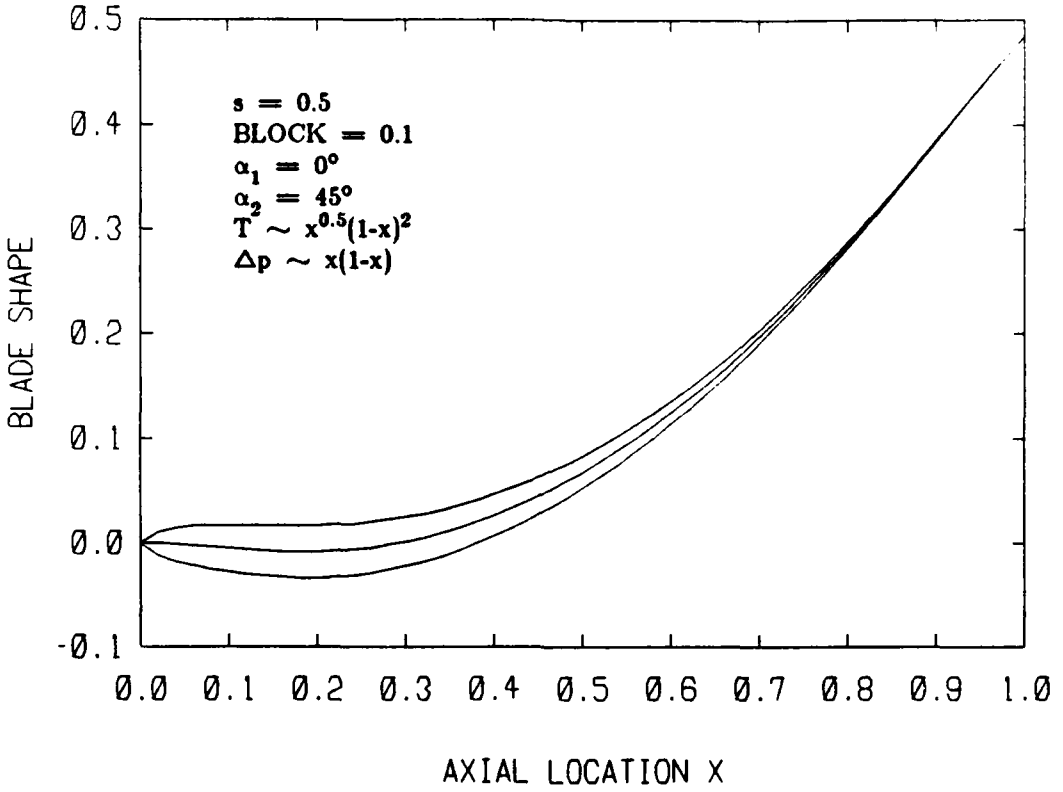


Figure 11 : Effects of maximum blockage on Cp's (BLOCK = 0.25)



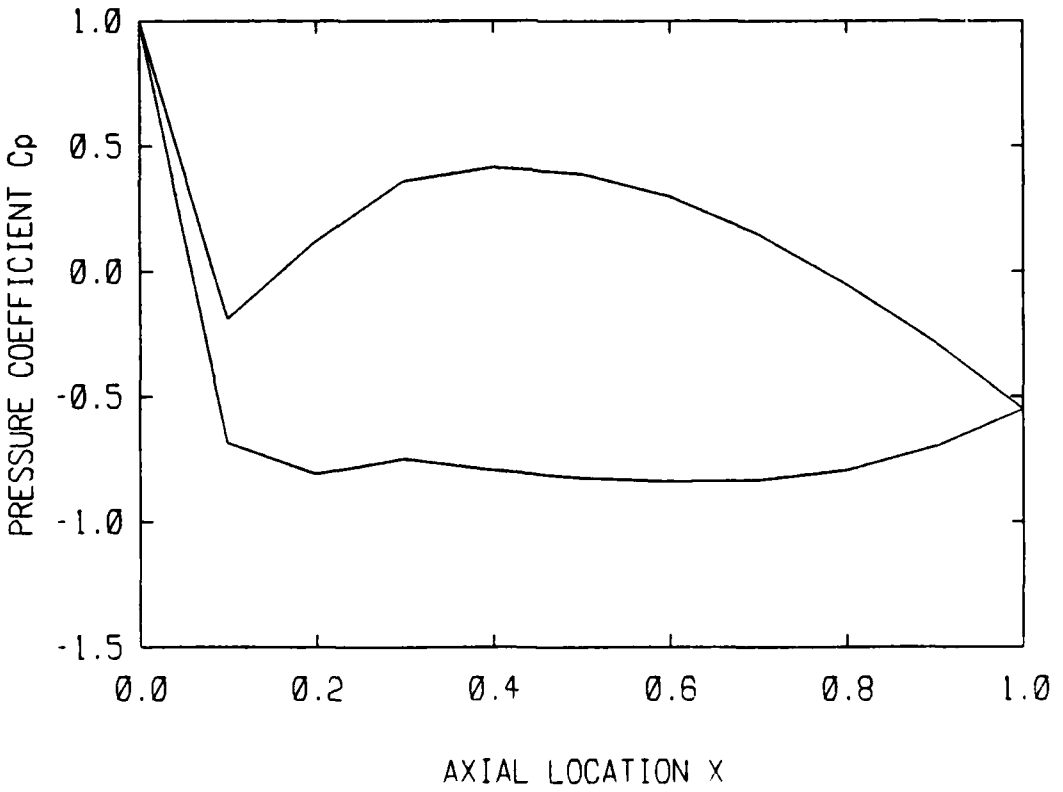


Figure 12 : Rounded leading edge inlet guide vane

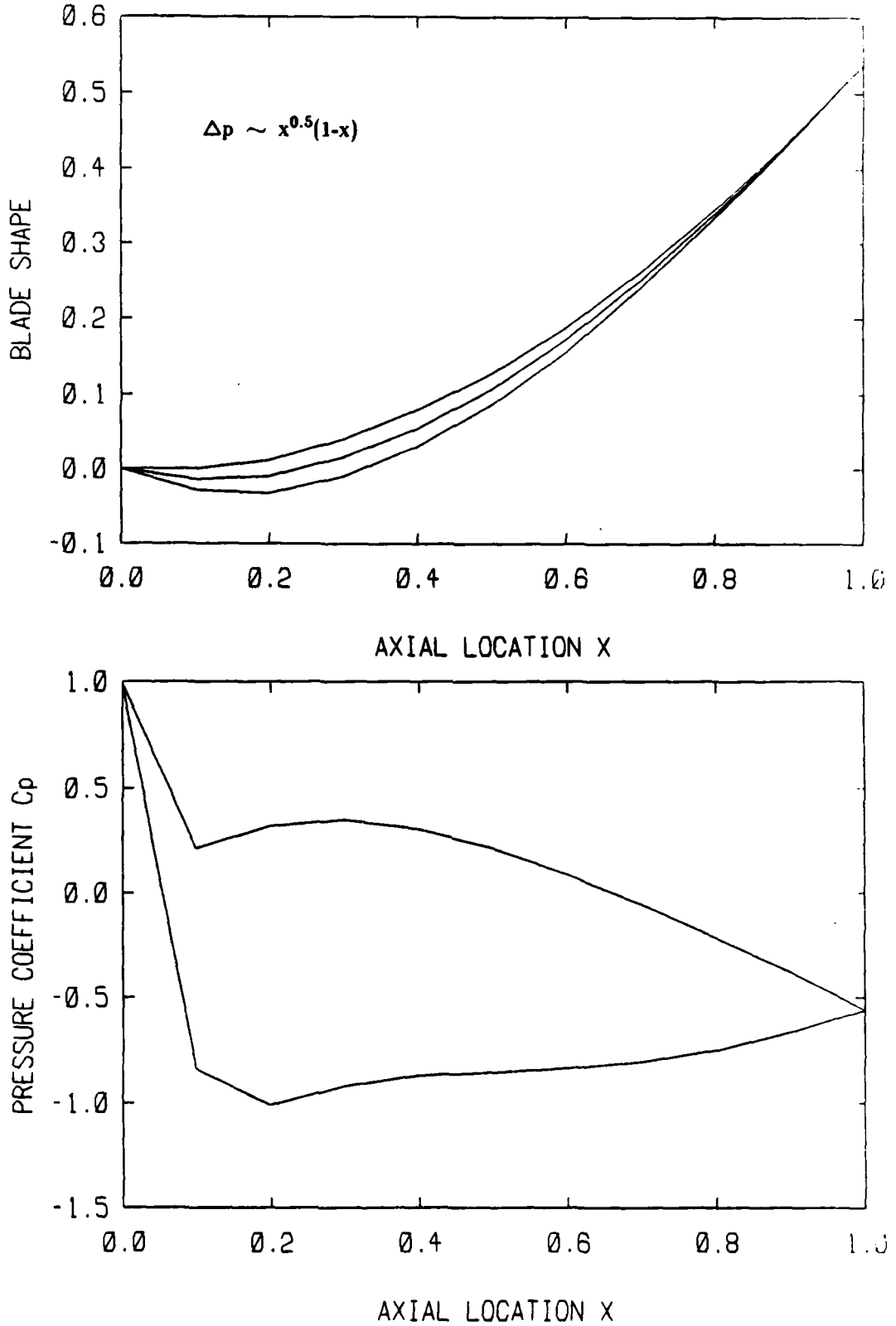


Figure 13 : Effects of loading distribution on Cp's (maximum loading at x = 1/3)

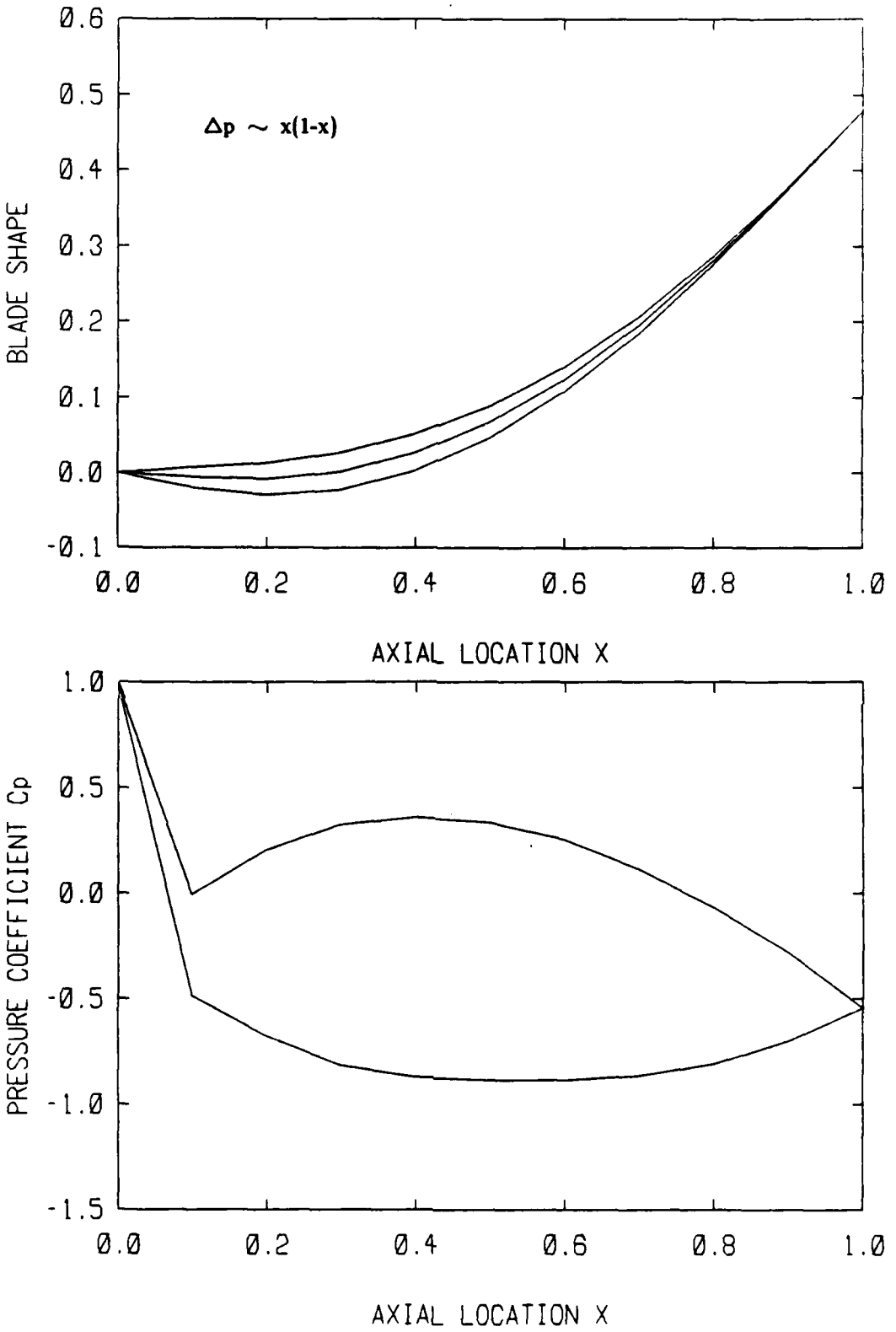


Figure 14 : Effects of loading distribution on Cp's (maximum loading at x = 1/2)

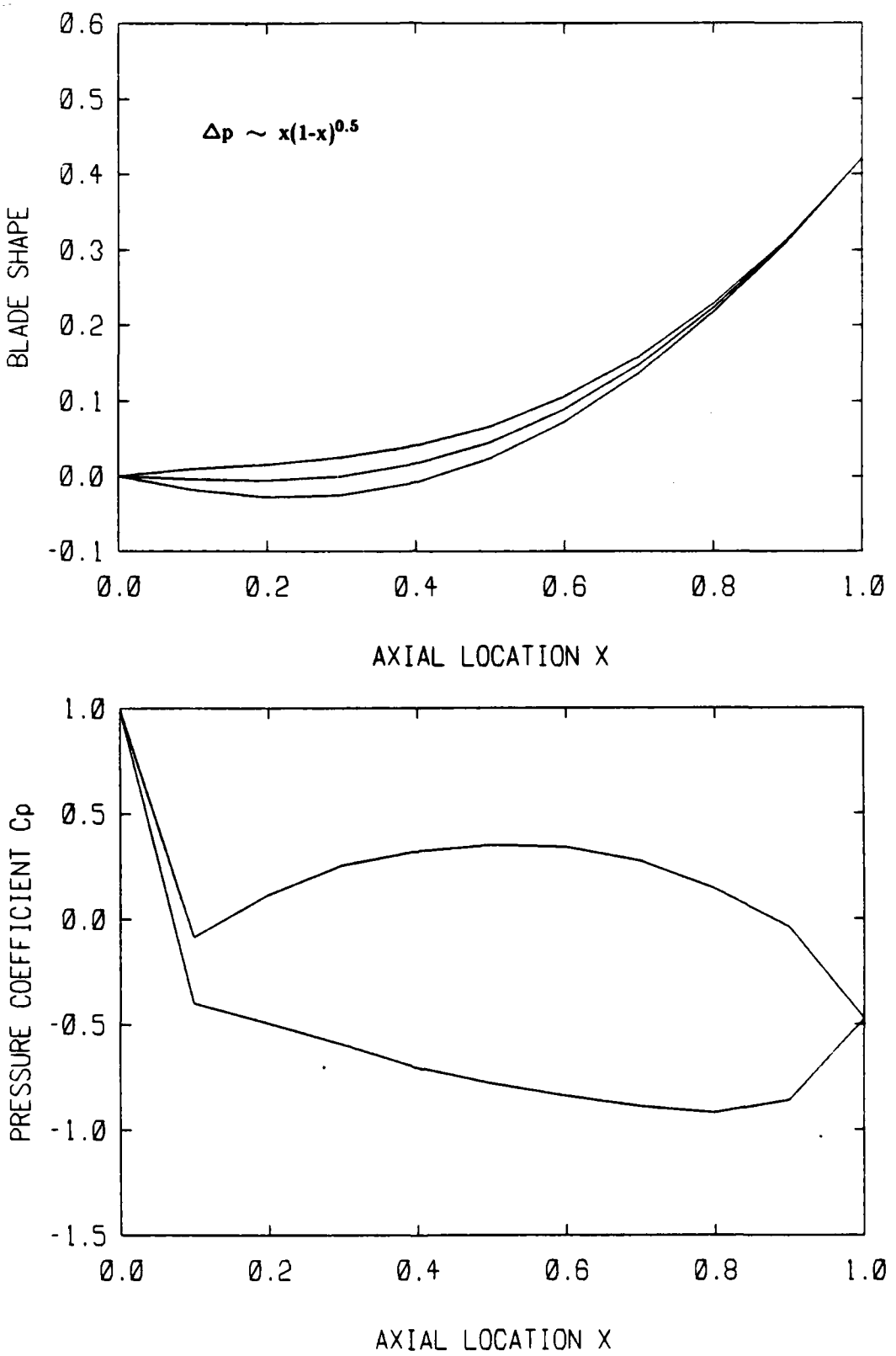


Figure 15 : Effects of loading distribution on Cp's (maximum loading at x = 2/3)

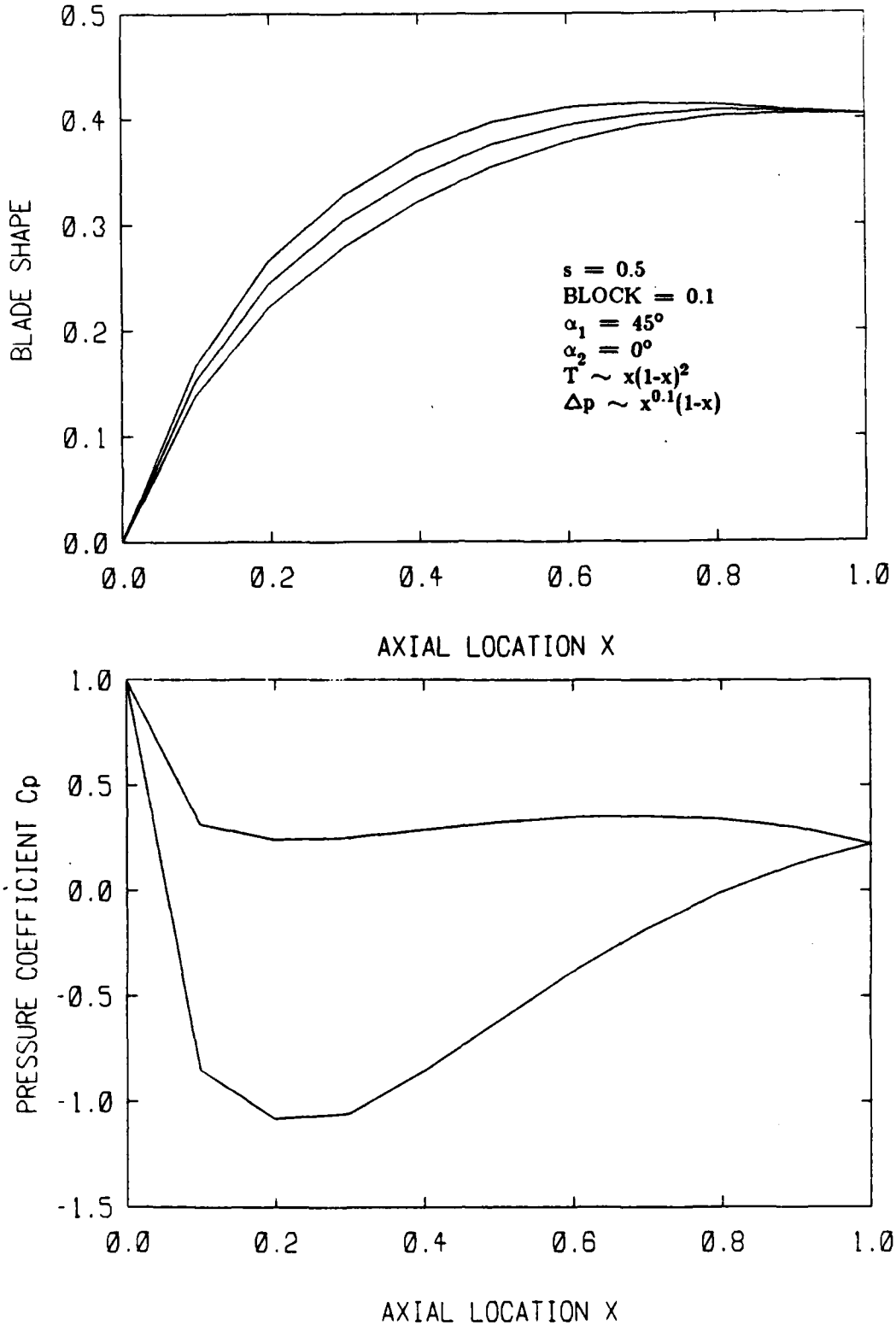


Figure 16 : Compressor blade

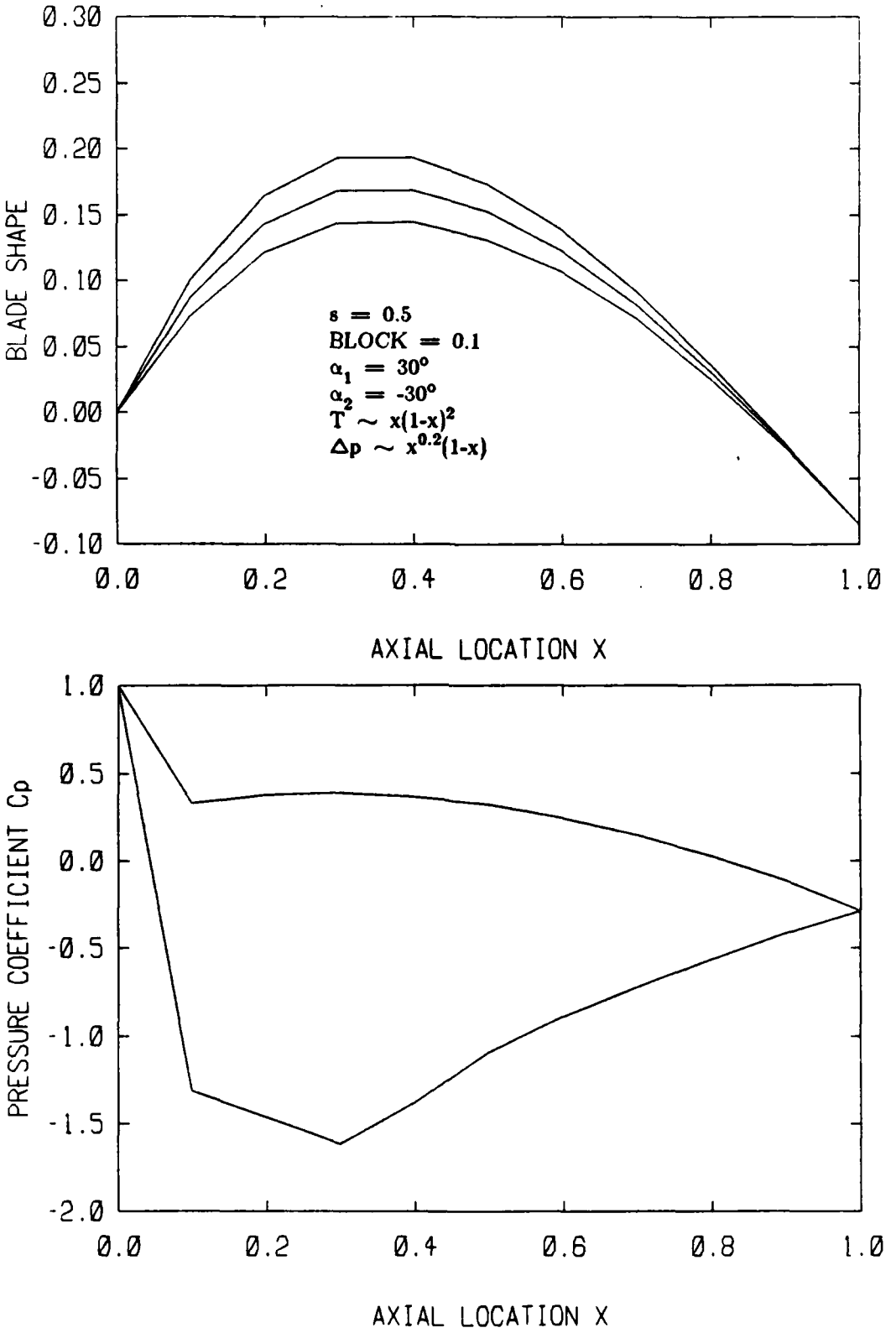
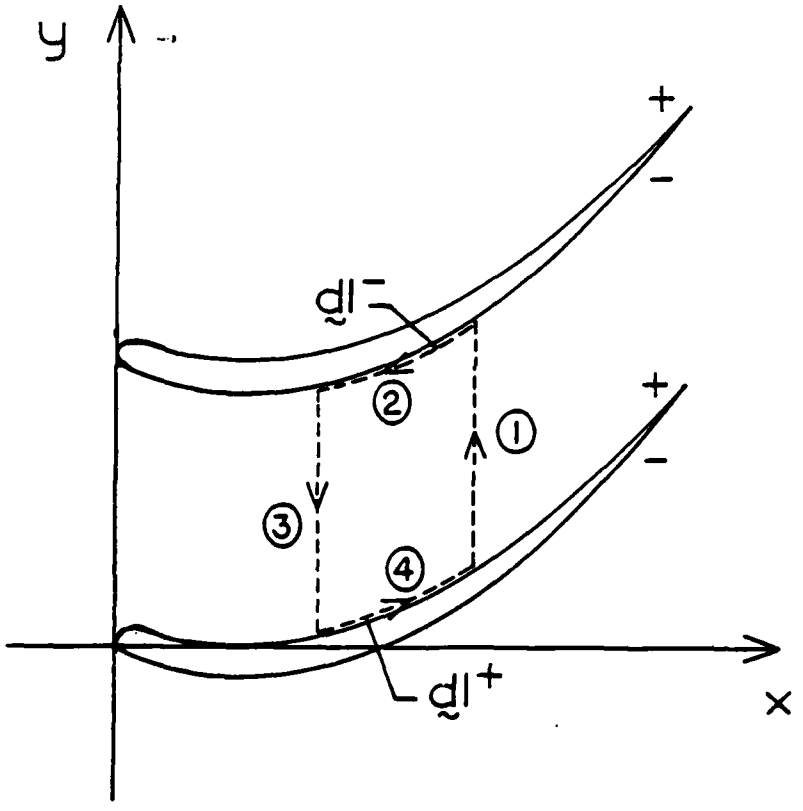


Figure 17 : Impulse blade

Appendix A : The Relationship between the Swirl Schedule and the Pressure Difference across the Blade

Consider the cascade geometry shown below



Under the assumption of incompressible, inviscid and uniform inlet flow condition, Bernoulli's equation is valid everywhere. We can write:

$$P^+ + \frac{1}{2}\rho(V^+)^2 = P^- + \frac{1}{2}\rho(V^-)^2$$

Therefore

$$\Delta P \equiv P^+ - P^- \propto (V^- + V^+)(V^- - V^+) \qquad \text{(A-1)}$$

We can do the following approximation:

$$(V^- + V^+) \sim 2\bar{V}_T$$

To estimate $(V^- - V^+)$, we consider the circulation along the closed path ①-②-③-④ shown in the figure. From Kelvin's theorem, we can write

$$\Gamma_c \equiv \Gamma_1 + \Gamma_2 + \Gamma_3 + \Gamma_4 = 0 \qquad \text{(A-2)}$$

Now,

$$\Gamma_1 = \int_{f+T}^{f+\Delta-T} V_y(x+\Delta x, y)\, dy = (\Delta - 2T)\, \bar{V}_{Ty}(x+\Delta x)$$

$$\Gamma_2 = -\underset{\sim}{V}^- \cdot d\underset{\sim}{\ell}^- \simeq -V^- \Delta x$$

$$\Gamma_3 = -\int_{f+T}^{f+\Delta-T} V_y(x,y)\, dy = -(\Delta - 2T)\, \bar{V}_{Ty}(x)$$

$$\Gamma_4 = \underset{\sim}{V}^+ \cdot d\underset{\sim}{\ell}^+ \simeq V^+ \Delta x$$

where

$$|d\underset{\sim}{\ell}^{\pm}| = \Delta x \sqrt{1 + (f' \pm T')^2} \simeq \Delta x$$

Substitute the above relations into (A-2), we obtain

$$(V^- - V^+)\, \Delta x \simeq (\Delta - 2T)\left[\bar{V}_{Ty}(x+\Delta x) - \bar{V}_{Ty}(x)\right]$$

and thus, in the limit of $\Delta x \to 0$

$$(V^- - V^+) \simeq \bar{V}'_{Ty}.$$

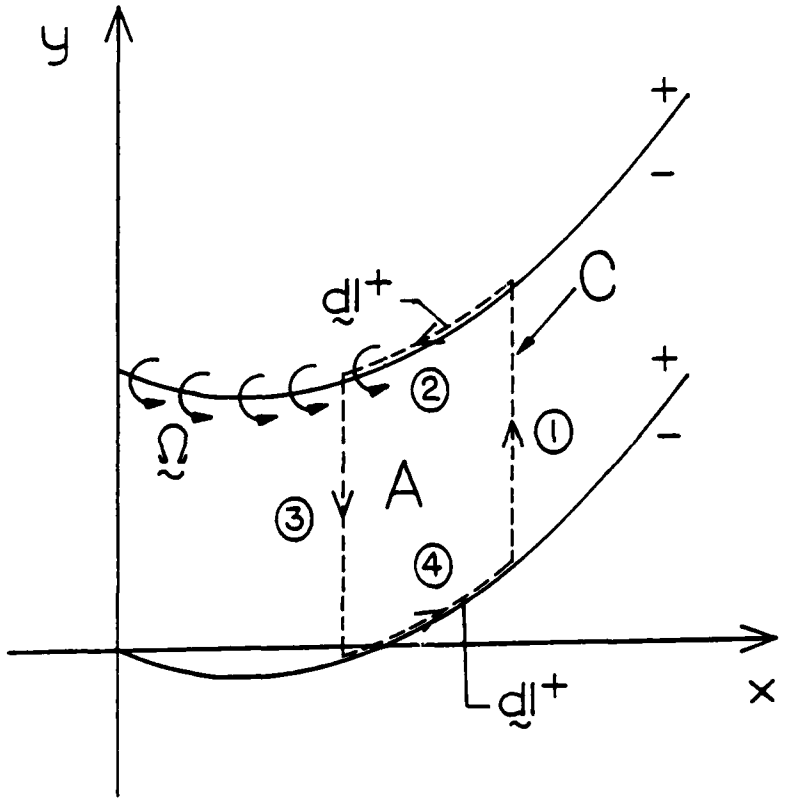
Therefore, equation (A-1) reduces to

$$\Delta p \propto \bar{V}_T\, (\bar{V}'_{Ty})$$

and we conclude that the pressure difference across the blade is directly proportional to the swirl schedule $\bar{V}'_{Ty}$. We will call $\bar{V}'_{Ty}$ the swirl schedule or the loading distribution.

Appendix B : The Bound Vorticity

In this appendix, we will show the relationship between the bound vorticity (vortices distributed on the blade camber lines to model their presence) and the swirl schedule (or the gradient of the pitch average velocity defined in chapter 2).



The flow is assumed to be incompressible and inviscid, and the far upstream flow is assumed to be uniform. The flow is thus irrotational and the vorticity must lie on the blade camber lines (see figure above). Therfore, we require:

$$\underset{\sim}{\Omega} \cdot \nabla \kappa = 0 \qquad \text{(B-1)}$$

Moreover, by vector identity

$$\nabla \cdot \underset{\sim}{\Omega} = 0 \tag{B-2}$$

To satisfy both equations (B-1) and (B-2) and the conditon that the vorticity direction must be normal to the x-y plane (2D assumption), we can write the vorticity field as

$$\underset{\sim}{\Omega} = s\,\delta_p(\alpha)\,(\nabla\alpha \times \nabla G) \tag{B-3}$$

where $\delta_p(\alpha)$ is the "periodic delta" function defined in Appendix C.

To find out what G is, consider the circulation around path C shown in figure. we may write, by Stokes theorem,

$$\oint_C \underset{\sim}{V} \cdot d\underset{\sim}{\ell} = \iint_A \underset{\sim}{\Omega} \cdot d\underset{\sim}{A} \tag{B-4}$$

Note that the line integrals along path ② and ④ cancel out each other exactly, and by substituting equation (B-3) into the right hand side of equation (B-4), along with the definition of the pitch average velocity, we can show that

$$\int_f^{f+s} V_y(x+\Delta x, y)\,dy - \int_f^{f+s} V_y(x,y)\,dy = \Delta x \int_f^{f+s} s\,\delta_p(\alpha)\left(-\frac{dG}{dx}\right) dy$$

and finally, by taking the limit as $\Delta x \to 0$, G in equation (B-3) is defined as

$$G = -\bar{V}_y$$

Appendix C : The Periodic Generalized Functions

The "periodic delta" function, the "sawtooth" function and the "smoothing" functions are constructed in this appendix.

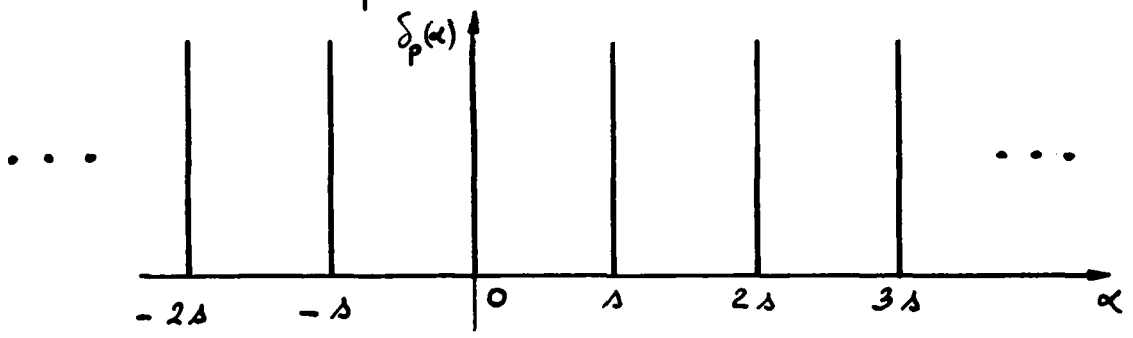
1. The "periodic delta" function

The "periodic delta" function may be expressed in a Fourier series of the form

$$\delta_p(\alpha) = \frac{1}{\Delta} \sum_{n=-\infty}^{\infty} e^{i\left(\frac{2\pi n}{\Delta}\right)\alpha}$$

where Δ is the spacing between blade camber lines, and α represents the blade camber's surfaces.

The plot of δ_p vs α looks like



It has the property

$$\int_{(n-1)\Delta}^{n\Delta} \delta_p(\alpha)\, d\alpha = 1.0$$

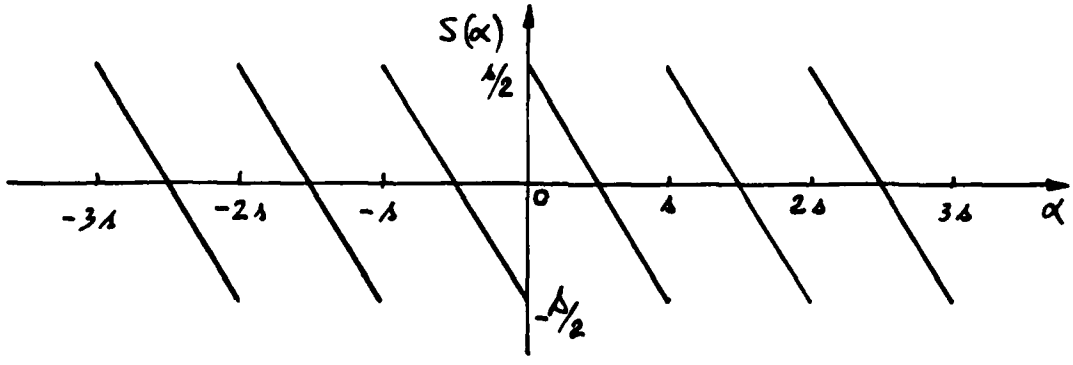
for any integer n .

2. The "sawtooth" function

The "sawtooth" function may be expressed in a Fourier series of the form

$$S(\alpha) = \sum_{n \neq 0} \frac{e^{i\left(\frac{2\pi n}{\Delta}\right)\alpha}}{i\left(\frac{2\pi n}{\Delta}\right)}$$

The plot of S vs α looks like



Its properties are:

- It has first derivative related to the "periodic delta" function by

$$S'(\alpha) = [\Delta \delta_p(\alpha) - 1]$$

- It has zero average between α lines

$$\int_{(n-1)\Delta}^{n\Delta} S(\alpha)\, d\alpha = 0$$

- It has a jump in magnitude of Δ everytime an α surface is crossed

- It reduces to a polynomial form when $(n-1)\Delta < \alpha < n\Delta$ for any integer n. Its polynomial form being

$$S(\alpha) = \frac{\Delta}{2} - \alpha$$

3. The "smoothing" functions

The "smoothing" functions $I_k(\alpha)$ are defined as

$$I_k(\alpha) = \sum_{n \neq 0} \frac{e^{i\lambda_n \alpha}}{(i\lambda_n)^{k+1}} \qquad k = 1, 2, 3, \ldots$$

where

$$\lambda_n = \frac{2\pi n}{\delta}$$

Their properties are:

- they have derivatives of the forms

$$\nabla I_k(\alpha) = I_{k-1}(\alpha)\, \nabla\alpha$$

- they have zero average between α lines

$$\int_{(n-1)\delta}^{n\delta} I_k(\alpha)\, d\alpha = 0$$

- they have polynomial forms in the intervals $(n-1)\delta < \alpha < n\delta$ for any integer n.

$$I_1(\alpha) \equiv I(\alpha) = -\frac{\delta^2}{12} + \frac{\delta\alpha}{2} - \frac{\alpha^2}{2}$$

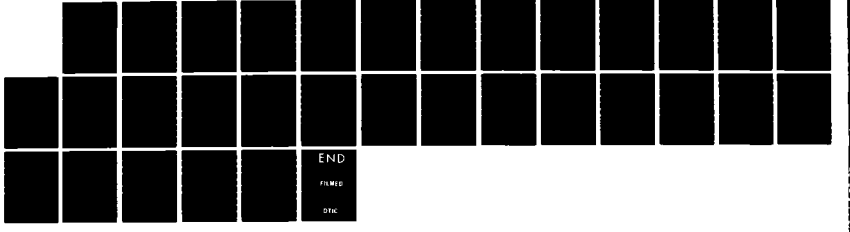
$$I_2(\alpha) \equiv J(\alpha) = -\frac{\delta^2\alpha}{12} + \frac{\delta\alpha^2}{4} - \frac{\alpha^3}{6}$$

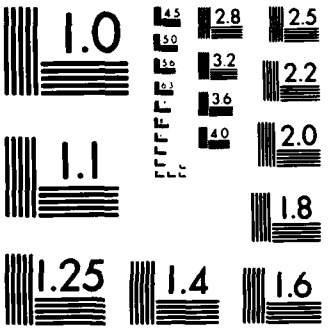
$$I_3(\alpha) \equiv K(\alpha) = -\frac{\delta^4}{720} - \frac{\delta^2\alpha^2}{24} + \frac{\delta\alpha^3}{12} - \frac{\alpha^4}{24}$$

$$\vdots$$

AD-A150 840 A TWO-DIMENSIONAL DESIGN METHOD FOR HIGHLY-LOADED BLADES IN TURBOMACHINES. (U) MASSACHUSETTS INST OF TECH CAMBRIDGE GAS TURBINE AND PLASMA D.. T Q DANG ET AL. 2/2

UNCLASSIFIED APR 83 GT/PDL-173 AFOSR-TR-85-0066 F/G 20/4 NL





Appendix D : The "Physical" Representations of β and δ

In this appendix, we will show the "physical" representations of β and δ. We will also find the strength of the vortices, sources and sinks which model the presence of the blades.

Consider the pitch average velocity $\bar{\underset{\sim}{V}}$ of the 'Finite-Thickness' problem. By definition

$$\bar{\underset{\sim}{V}}(x) = \frac{1}{\Lambda}\int_{f}^{f+\Lambda} \underset{\sim}{V}_2(x,y)\,dy$$

using equation (4.2-16) for $\underset{\sim}{V}_2$, $\bar{\underset{\sim}{V}}$ can be shown to be

$$\bar{\underset{\sim}{V}} = (\bar{V}_{Tx} + \beta)\,\hat{e}_x + (\bar{V}_{Ty} - \delta)\,\hat{e}_y \qquad \text{(D-1)}$$

Since $\bar{V}_{Tx}$ and $\bar{V}_{Ty}$ are the gap average velocity components, β and $-\delta$ represent the x-component and y-component of the "imaginary" flow in the "blade" region respectively. We expect these variables to be proportional to the blockage distribution.

Consider the curl of $\bar{\underset{\sim}{V}}$

$$\nabla \times \bar{\underset{\sim}{V}} = (V'_{Ty} - \delta') \qquad \text{(D-2)}$$

In chapter 2, we have shown that if the vorticity is distributed along the blade camber, then its strength is related to the gradient of the y-component of the pitch average velocity. Equation (D-2) is indeed the case.

Consider the divergence of

$$\nabla \cdot \bar{\underset{\sim}{V}} = (\bar{V}'_{Tx} + \beta') \qquad \text{(D-3)}$$

Since the flow is incompressible, $(\bar{V}'_{Tx} + \beta')$ represents the source/sink distribution. Note that the boundary conditions $(\bar{V}'_{Tx} + \beta') = 0$ (equation(4.2-17)) at $X = 0$ and $X = 1$ are the conditions required for the blade profile to close there.

Appendix E : Numerical Difficulties

In this appendix, we will attempt to understand the convergence problem encountered in the iteration process for the blade camber line f when "partial smoothing" is used. We study the mathematical behavior of β and δ.

By combining A_T, B_T, β and δ in the "partial smoothing" forms of equations (4.3-1) and (4.3-2), we arrive at the following boundary value problems:

$$F(x)(f'^4-1)X'' + F(x)f'f''(3-f'^2)X' + X = \bar{V}_{Tx} - \left(\frac{\overline{\partial \varphi_{hc}}}{\partial x}\right)_T + F(x)\left[2f'(1+f'^2)Y'' + f''(3-f'^2)Y'\right] \quad \text{(E-1)}$$

with boundary conditions

$$X'(0) = 0$$

$$X'(1) = 0$$

and

$$F(x)(f'^4-1)Y'' + F(x)f'f''(3-f'^2)Y' + Y = -\bar{V}_{Ty} + \left(\frac{\partial \varphi_{hc}}{\partial y}\right)_T + F(x)\left[-2f'(1+f'^2)X'' - f''(3-f'^2)X'\right] \quad \text{(E-2)}$$

with boundary conditions

$$Y'(0) = 0$$

$$Y'(1) = 0$$

where

$$X \equiv \bar{V}_{Tx} + \beta$$

$$\gamma \equiv -\bar{V}_T + \zeta$$

$$F(x) \equiv \frac{J(\tau)}{S(\tau)} \left(\frac{1}{1+f'^2} \right)^3$$

Since f itself is a function of β and ζ, we have a boundary value problem consisting of two coupled non-linear second order ordinary differential equations. In solving them iteratively, we expect that certain difficulties can arise.

The strut problem

We study the mathematical behavior of β and ζ by first consider the case where f is identically zero everywhere, but T is finite. This is the case of a symmetric blade (or a strut). In this case, γ is also identically zero everywhere, and the above boundary value problem reduces to

$$-\frac{J(\tau)}{S(\tau)} \beta'' + \beta = \frac{J(\tau)}{S(\tau)} \bar{V}_{Tx}'' - \left(\overline{\frac{\partial \varphi_{h2}}{\partial x}} \right)_T \qquad \text{(E-3)}$$

with boundary conditions

$$\beta'(0) = -\bar{V}_{Tx}'(0)$$

$$\beta'(1) = -\bar{V}_{Tx}'(1)$$

Consider the homogeneous solution of the above differential equation by assuming β to be of the form

$$\beta = e^{i E(x)}$$

Substituting it into equation (E-3), we obtain

$$-\frac{J(\tau)}{S(\tau)} (i E'' - E'^2) + 1 = 0$$

Assume $E'' << E'^2$. Then

$$E(x) = \pm \int \sqrt{-\frac{S(T)}{J(T)}}\, dx$$

Consider the case where $T/\Delta << 1$. Then, by definition

$$S(T) \sim \mathcal{O}(\Delta)$$

$$J(T) \sim \mathcal{O}(-\Delta^2 T)$$

and thus

$$-\frac{J(T)}{S(T)} \sim \mathcal{O}\left(\frac{1}{\Delta T}\right)$$

It can be shown that under the assumption of $T/\Delta << 1$, we can say that $E''/E'^2 \sim \mathcal{O}(T)$ and the assumption $E'' \ll E'^2$ is justifiable, except perhaps near $X = 0$ or 1.

Finally, we can write

$$\beta \sim e^{\pm i \int \sqrt{\frac{1}{\Delta T}}\, dx}$$

As an example, let

$$T(x) = 2\Delta\,(BLOCK)\, x(1-x)$$

where BLOCK is defined in subsection 4.5.1 . Then, it can be shown that

$$\beta \sim e^{\pm i \sqrt{\frac{2}{\Delta^2 BLOCK}}\,(\sin^{-1}\sqrt{x})}$$

With the assumption that $T/\Delta << 1$, we conclude that β can be a highly oscillating function, having a "natural frequency" of the order of $\mathcal{O}\left(\sqrt{\frac{2}{\Delta^2 BLOCK}}\right)$.

Finally, we look at the amplitude of β . By definition, i.e. equation (4.2.13) in "partial smoothing" form

$$\beta \equiv \frac{J(\tau)}{S(\tau)} \left(A'_\tau - B_\tau f' \right) - \left(\overline{\frac{\partial \varphi_{h_2}}{\partial x}} \right)_\tau$$

Therefore, the above analysis shows that β behaves like

$$\beta \sim \mathcal{O}(\Delta T)$$

and thus, the source/sink distribution ($\bar{V}'_{Tx} + \beta'$) oscillates.

A computer program was written to solve the differential equation (E-3) using the Chebyshev collocation technique [7], [8]. Numerical results show that β oscillates with a natural frequency in agreement with the above analysis.

In classical aerodynamics we know that, for a smooth blade with finite thickness, the source/sink distribution used to model its presence should exhibit like a "sine" wave with a singular point at the leading edge. Numerically, in order to resolve this singular point, we would need an infinite number of terms in the smoothing series. We conclude that in the case of "partial smoothing" (by using only two terms in the smoothing series), we are unable to represent the usual source/sink distribution. However, such a representation is also not necessary.

Accordingly, we will call ($\bar{V}'_{Tx} + \beta'$) the "modified" source/sink distribution, and show that it can be used to produce corresponding blade shapes satisfactorily. Thus we seek a practical method for the design problem, using "partial smoothing", without necessarily having to go into more extensive mathematical development.

The loaded blade problem

From the above discussion of the strut problem, we expect expect both β and δ to have oscillating behaviors when the blades are loaded. Two iteration schemes for f were investigated using "partial smoothing".

We note that equations (E-1) and (E-2) have certain symmetry. Rewrite them in operator forms:

$$\begin{aligned} \mathcal{L}_1\{X\} - \mathcal{L}_2\{Y\} &= \mathcal{S}_1 \\ \mathcal{L}_1\{Y\} + \mathcal{L}_2\{X\} &= \mathcal{S}_2 \end{aligned} \qquad \text{(E-4)}$$

where

$$\mathcal{L}_1\{\ \} \equiv \left[F(x)(f'^4-1)\frac{d^2}{dx^2} + F(x)f'f''(3-f'^2)\frac{d}{dx} + 1\right]\{\ \}$$

$$\mathcal{L}_2\{\ \} \equiv \left[2f'(1+f'^2)\frac{d^2}{dx^2} + f''(1-3f'^2)\frac{d}{dx}\right]\{\ \}$$

$$\mathcal{S}_1 \equiv \bar{V}_{Tx} - \left(\overline{\frac{\partial \varphi_{nl}}{\partial x}}\right)_T$$

$$\mathcal{S}_2 \equiv -\bar{V}_{Ty} + \left(\overline{\frac{\partial \varphi_{nl}}{\partial y}}\right)_T$$

The boundary conditions remain the same as in equations (E-1) and (E-2).

Method 1

In this method, we attempt to solve equations (E-4) simultaneously for β and δ using the Chebyshev collocation technique.

We express β and δ as Chebyshev series and convert equations (E-4) into matrix forms:

$$[A]\{X\} - [B]\{Y\} = \{S_1\}$$

$$[A]\{Y\} + [B]\{X\} = \{S_2\}$$

which can be arranged in the form

$$\left[\begin{array}{c|c} [A] & -[B] \\ \hline [B] & [A] \end{array}\right] \left\{\begin{array}{c} \{X\} \\ \hline \{Y\} \end{array}\right\} = \left\{\begin{array}{c} \{S_1\} \\ \hline \{S_2\} \end{array}\right\}$$

X and Y are solved by inverting the above matrix using an IMSL subroutine [9].

Results show that when more than approximately 11 collocation points are used in the calculation, convergence in f cannot be achieved. β and δ are found to oscillate and their Chebyshev coefficients fail to converge. When around 51 points or more collocation points are used in the calculation, the iteration process diverges rapidly. Two conclusions can be made from the results of this method:

1. β and δ pocess oscillating behavior as predicted by the analysis of the strut problem. By using more than 51 collocation points, the numerical calculation tries to resolve the Gibbs phenomenon at the leading edge, but fails to do so because of "partial smoothing".

2. The iteration process can diverge rapidly because of the very nature of the iteration process. We note that even though we are solving X and Y simultaneously, we are in effect solving equation (E-4) iteratively because f and its derivatives are

updated at every iteration. We can therefore look at the iteration process of method 1 as if we were attempting to solve for X and Y in the following manner:

a. First update

$$\mathcal{L}_1 \{ X^{n+1} \} = \mathcal{L}_2 \{ Y^n \} + S_1^n$$

b. Then update

$$\mathcal{L}_1 \{ Y^{n+1} \} = - \mathcal{L}_2 \{ X^n \} + S_2^n$$

where n is the iteration level in the iteration process for f .

Since operators $\mathcal{L}_1$ and $\mathcal{L}_2$ are both expected to have normal mode solutions of oscillating behavior, we conclude that there is a chance for a driven reasonance to occur during the iteration process for f , which can lead to divergence of the iteration scheme itself. We think that this is indeed the case. When too many terms are kept in the Chebyshev series, higher modes are present resulting in a greater chance for reasonance to occur. When fewer terms (around 10) are kept in the Chebyshev series, we are staying away from the natural frequency of the operators $\mathcal{L}_1$ and $\mathcal{L}_2$ resulting in a stable iteration process.

Method 2

In this method, we use the iteration process described in section 4.3 . Derivatives are computed numerically using two methods: Chebyshev collocation method, and finite difference method. Numerical results show that similar problems as in method 1 were encountered here. This should be expected, as observed above, because iterations are being used.

We decided to use method 2 for our design method because of two reasons

1. method 2 is much more efficient than method 1 (faster, cheaper and simpler). In method 1, we are required to invert a matrix at each iteration of f . In method 2, we are required to compute derivatives instead.

2. method 2 can easily be modified if we wish to keep more terms in the smoothing series. Equations (E-1) and (E-2) are only valid when the first two terms in the smoothing series are kept.

Finally, the finite difference scheme (central difference) is used to compute derivatives because it is numerically more stable than the Chebyshev collocation method.

A computer program was written using the above method. It is found that when around 11 points are used in the calculations, convergence in f is achieved in about 10 iterations. When more than around 20 points are used in the calculation, f fails to converge. In order to resolve this problem, we propose to use a filter. The calculation procedure is:

1. when more than 11 points are used in the calculation, iterate for f using a "filter".

2. when 11 points or less are used in the calculation, iterate for f without using the "filter".

Filtering method

Two different "filters" are developed for the above iteration scheme:

1. f is filtered using a least-squares chord-wise fitting method [10]. The combination $(G_\tau' + \delta' - A_\tau f')$ in equation (4.2-22) is filtered using a fourth order polynomial. The motivation for using a polynomial curve fitting method is that we expect f itself be represented by a polynomial of low order.

2. the pressure coefficients are filtered by taking the average of the maximum and minimum envelopes of the C_p curves. The

C_p curves are filtered only near the leading edge region ($0 < x < .4$). The envelopes are constructed by straight lines going through the maximum and minimum points of the C_p curves. This procedure has been chosen to date for its simplicity, it clearly admits improvement possibilities near the leading edge. But, till now at least, this approach has compared adequately with known results (see Text).

Appendix F : Computer Code of "Zero-Thickness" problem

```
C ********************************************************
C *                                                      *
C *  PROGRAM NAME   THIN FOR                             *
C *  MAIN PROGRAM FOR INVERSE DESIGN OF COMPRESSOR BLADES *
C *  2-D INCOMPRESSIBLE, INVISCID, INFINITELY THIN   THE  *
C *  LOADING DISTRIBUTION IS OF PARABOLIC FORM           *
C *                                                      *
C *  INPUTS      S - SPACING                             *
C *            ALP1 - INLET ANGLE (DEGREE)               *
C *            ALP2 - OUTLET ANGLE (DEGREE)              *
C *             IJK - NUMBER OF POINTS                   *
C *            ITER - MAXIMUM NUMBER OF ITERATIONS ALLOWED *
C *             ERR - CONVERGENCE CRITERIA 'ERROR'       *
C *                                                      *
C ********************************************************
      REAL X(101),VMY(101),DVMY(101),DDVMY(101),FM(101),A(101)
     1,DA(101),FNEW(101),F(101),DF(101),DDF(101),PLOT(3,101)
      COMMON/S,A0,A1,B0,B1,PI,IJK,X,NMAX,COR
C
C READ STATEMENT
C
      READ(1,*)S,ALP1,ALP2,IJK,ITER,ERR
      WRITE(2,50)S,ALP1,ALP2,IJK
50    FORMAT(5X,'SPACING S = ',F6 3/5X,'INLET ANGLE ALP1 = ',F7 3/
      15X,'OUTLET ANGLE ALP2 = ',F7 3/5X,'NUMBER OF POINTS IJK = '
     1,I3///)
C
C INITIALIZE VARIABLES FOR CALCULATION PURPOSE
C
      NMAX=20
      PI=3 141592654
      RAD=57 29577951
      TAN1=TAN( 017453294*ALP1)
      TAN2=TAN( 017453294*ALP2)
      XIJK=IJK-1
      DX=1 /XIJK
      XX=1 +DX
      SUM=0
      DO 5 N=1,IJK
      XN=N
      SUM=SUM+(1 /(XN*XN))
5     CONTINUE
      COR=1 6449341/SUM
C
C COMPUTE COMPUTATIONAL LOCATIONS
C
      DO 10 I=1,IJK
      J=IJK+1-I
      XX=XX-DX
      X(I)=XX
      PLOT(3,J)=X(I)
10    CONTINUE
C
C COMPUTE INPUT FOR PARABOLIC LOADING CASE
C
      DO 15 I=1,IJK
      J=IJK+1-I
      XX=X(I)
      CONST=6 *(TAN2-TAN1)
      VMY(I)=CONST*( 5*XX*XX-XX*XX*XX/3 )+TAN1
```

```
      DVMY(I)=CONST*XX*(1 -XX)
      DDVMY(I)=CONST*(1 -2 *XX)
      FM(I)=CONST*(XX**3/6 -XX**4/12 )+TAN1*XX
      PLOT(1,J)=FM(I)
      DF(I)=VMY(I)
      DDF(I)=DDVMY(I)
      DEN=1 +DF(I)*DF(I)
      F(I)=FM(I)+ 08333333*S*S*DVMY(I)*(-1 +DF(I)*DF(I)/DEN)
15    CONTINUE
      FNEW(IJK)=0
      F(IJK)=0
C
C START ITERATION PROCESS FOR CAMBER LINE
C
      OLDERR=100
      DO 1 NIT=1,ITER
C
C COMPUTE A AND ITS DERIVATIVE
C
      DO 100 I=1,IJK-1
      DEN=1 +DF(I)*DF(I)
      A(I)=-DF(I)*DVMY(I)/DEN
      TERM1=-DEN*(DDF(I)*DVMY(I)+DF(I)*DDVMY(I))
      TERM2=2 *DF(I)*DF(I)*DDF(I)*DVMY(I)
      DA(I)=(TERM1+TERM2)/DEN
100   CONTINUE
C
C COMPUTE EDGE VALUES
C
      DEN0=1 +DF(IJK)**2
      DEN1=1 +DF(1)**2
      B0=DDVMY(IJK)*(1 -2 *DF(IJK)*DF(IJK)/DEN0)/DEN0
      B1=DDVMY(1)*(1 -2 *DF(1)*DF(1)/DEN1)/DEN1
      A0=-2 *DF(IJK)*DDVMY(IJK)*(1 -DF(IJK)*DF(IJK)/DEN0)/DEN0
      A1=-2 *DF(1)*DDVMY(1)*(1 -DF(1)*DF(1)/DEN1)/DEN1
C
C UPDATE CAMBER LINE F(X)
C
      XNORM=-2 *FSUM(IJK,F)/S
      ERRMAX=0
      SUMERR=0
      XLOC=-1
      DO 101 I=1,IJK
      DF(I)=VMY(I)+ 08333333*S*S*(-DDVMY(I)-A(I)*DDF(I)-DA(I)*DF(I))
      1-2 *DFSUM(I,F,DF)/S
101   CONTINUE
      DO 102 I=1,IJK-1
      FNEW(I)=FM(I)+ 08333333*S*S*(-DVMY(I)-A(I)*DF(I))-2 *FSUM(I,F)/S
      1-XNORM
      ERROR=ABS(FNEW(I)-F(I))
      SUMERR=SUMERR+ERROR
      IF(ERROR GT ERRMAX)XLOC=X(I)
      IF(ERROR GT ERRMAX)ERRMAX=ERROR
102   CONTINUE
C
C CHECK FOR CONVERGENCE IN F(X)
C
      WRITE(2,55)NIT,ERRMAX,XLOC
55    FORMAT(10X,'ITER #',I2,' ----- ERRMAX = ',F7 5
      1,'   AT X = ',F8 5)
```

```
      AVGERR=SUMERR/XIJK
      IF(ERRMAX GT (1 1*OLDERR))WRITE(2,54)AVGERR
54    FORMAT(//2X,'***** ITERATION SCHEME DIVERGES !!!! *****'/
     1       2X,'      AVERAGE ERROR = ',F9 5)
      IF(ERRMAX GT (1 1*OLDERR))GO TO 998
      IF(ERRMAX LE ERR)GO TO 998
C
C UPDATE VALUES FOR NEXT ITERATION
C
      DO 103 I=1,IJK-1
      F(I)=FNEW(I)
103   CONTINUE
      OLDERR=ERRMAX
C
1     CONTINUE
C
998   CONTINUE
C
C OUTPUT
C
      WRITE(2,60)
60    FORMAT(//T10,'X',T25,'FM',T37,'FLOW ANGLE',T55,'F',T67
     1,'BLADE ANGLE'/)
      DO 501 I=1,IJK
      J=IJK+1-I
      PLOT(2,I)=FNEW(J)
      ANGLFM=RAD*ATAN(VMY(J))
      ANGLF=RAD*ATAN(DF(J))
      WRITE(2,61)X(J),FM(J),ANGLFM,FNEW(J),ANGLF
61    FORMAT(5(5X,F10 5))
501   CONTINUE
C
C CALL JCF PLOTTING SUBROUTINE
C
      CALL QPICTR(PLOT,3,IJK,QY(1,2),QX(3),QLABEL(4)
     1,QYLAB('BLADE [#2] - MEAN STREAMLINE [#1]')
     1,QXLAB('AXIAL LOCATION X'))
C
      STOP
      END
C
C ***********************************************************
C
C COMPUTE FUNCTION FSUM
C
      FUNCTION FSUM(I,F)
      REAL F(101),X(101)
      COMMON/S,A0,A1,B0,B1,PI,IJK,X,WMAX,COR
      REAL LAMDM
      SUM=0
      DO 100 M=1,WMAX
      XM=M
      LAMDM=2 *PI*XM/S
      DELF0=(F(I)-F(IJK))*LAMDM
      DELF1=(F(I)-F(1))*LAMDM
      T0=EXP(-LAMDM*X(I))/LAMDM**3
      T1=EXP(-LAMDM*(1 -X(I)))/LAMDM**3
      TERM0=B0*COS(DELF0)+A0*SIN(DELF0)
      TERM1=-B1*COS(DELF1)+A1*SIN(DELF1)
      SUM=SUM+T0*TERM0+T1*TERM1
```

```
100     CONTINUE
        FSUM= 5*S*COR*SUM
        RETURN
        END
C
C ***********************************************************
C
C COMPUTE FUNCTION DFSUM
C
        FUNCTION DFSUM(I,F,DF)
        REAL F(101),DF(101),X(101)
        COMMON/S,A0,A1,B0,B1,PI,IJK,X,NMAX,COR
        REAL LAMDM
        SUM=0
        DO 100 M=1,NMAX
        XM=M
        LAMDM=2 *PI*XM/S
        DELF0=(F(I)-F(IJK))*LAMDM
        DELF1=(F(I)-F(1))*LAMDM
        COS0=COS(DELF0)
        COS1=COS(DELF1)
        SIN0=SIN(DELF0)
        SIN1=SIN(DELF1)
        EXP0=EXP(-LAMDM*X(I))/LAMDM**2
        EXP1=EXP(-LAMDM*(1 -X(I)))/LAMDM**2
        TERM00=-EXP0*(B0*COS0+A0*SIN0)
        TERM01=DF(I)*EXP0*(-B0*SIN0+A0*COS0)
        TERM10=EXP1*(-B1*COS1+A1*SIN1)
        TERM11=DF(I)*EXP1*(B1*SIN1+A1*COS1)
        SUM=SUM+TERM00+TERM01+TERM10+TERM11
100     CONTINUE
        DFSUM= 5*S*COR*SUM
        RETURN
        END
C
C ***********************************************************
```

Appendix G : Computer Code of "Finite-Thickness" problem

```
C ***********************************************************
C *                                                         *
C *  PROGRAM NAME   THICK FOR                               *
C *  MAIN PROGRAM FOR INVERSE DESIGN OF COMPRESSOR BLADES   *
C *  2-D INCOMPRESSIBLE, INVISCID WITH FINITE THICKNESS     *
C *  USING THE CENTRAL DIFFERENCE METHOD TO COMPUTE         *
C *  DERIVATIVES                                            *
C *                                                         *
C ***********************************************************
C
      REAL X(101),T(101),DT(101),DDT(101),RS(101),RI(101),RJ(101)
     1,VMX(101),DVMX(101),DDVMX(101),VMY(101),DVMY(101)
     1,DDVMY(101),FM(101),F(101),FOLD(101),DF(101),DDF(101)
     1,BETA(101),DBETA(101),DDBETA(101),DELT(101),DDELT(101)
     1,DDDELT(101),PHIHXB(101),PHIHYB(101),DPHIHX(101),DPHIHY(101)
     1,XLOAD(101),XILOAD(101),CPT(101),CPB(101)
     1,Y(101),VXT(101),VYT(101),VXB(101),VYB(101),PLOT(6,101)
     1,AAAI(10,10),OSC(101),SMOSC(101),SMCF(101),OPT(101)
     1,A(101),DA(101),DDA(101),B(101),DB(101)
      REAL LAMDM
      COMMON/COMM1/S,A0,A1,B0,B1,PI
      COMMON/COMM2/T,DT,DDT,RS,RI,RJ,DVMX,DVMY,DDVMX,DDVMY,FM
      READ(5,*)BLOCK,S,ALP1,ALP2,IJK,MMAX,ITER,ERR,CA,CB,CC,CD,NOPT
     1,IPOSER,BLADET,BLADEB,PRESST,PRESSB
      WRITE(1,1)BLOCK,S,ALP1,ALP2,IJK,ITER,ERR,NOPT,CA,CB,CC,CD
1     FORMAT(///5X,'INPUT PARAMETERS'//T5,'MAX  BLOCKAGE = ',
     1F10 5/T5,'SPACING = ',F10 5/T5,'INLET ANGLE = ',F10 5/
     1T5,'OUTLET ANGLE = ',F10 5//
     1T5,'NUMBER OF POINTS IJK = ',I3/
     1T5,'MAX  NUMBER OF ITERATIONS ALLOWED = ',I3/
     1T5,'MAX  ERROR IN F(X) ALLOWED   ERRMAX = ',F10 6/
     1T5,'FILTERING OPTION = ',I1//T7,'BLOCKAGE AND LOADING PARAMETERS'/
     1T9,'A = ',F5 2/T9,'B = ',F5 2/T9,'C = ',F5 2/T9,'D = ',F5 2//)
C
C INITIALIZE VARIABLES FOR COMPUTATION
C
      PI=3 141592654
      RALP1= 017453293*ALP1
      RALP2= 017453293*ALP2
      RAD=57 29577951
      TAN1=TAN(RALP1)
      TAN2=TAN(RALP2)
      VYN= 5*(TAN1+TAN2)
      VONSET=(1 +VYN**2)** 5
      AMPT=S*BLOCK/(2 *((CA/(CA+CB))**CA)*((CB/(CA+CB))**CB))
      XIJK=IJK-1
      DX=1 /XIJK
      XX=1 +DX
      SUM=0
      DO 666 N=1,MMAX
      XN=N
      SUM=SUM+(1 /(XN*XN))
666   CONTINUE
      COR=1 644934068/SUM
C
C COMPUTE LOCATION
C
      DO 20 I=1,IJK
      XX=XX-DX
      X(I)=XX
```

```
20      CONTINUE
        IF(NOPT EQ 1)CALL SMAI(IJK,IPOWER,X,AAAI)
C
C COMPUTE THICKNESS INPUT
C
        T(IJK)=0
        T(1)=0
        DO 21 I=2,IJK-1
        XX=X(I)
        T(I)=AMPT*(XX**CA)*(1 -XX)**CB
21      CONTINUE
        CALL DERIV(IJK,DX,T,DT)
        DT(1)=0
        CALL DERIV(IJK,DX,DT,DDT)
C
C COMPUTE INPUT LOADING
C
        XLOAD(IJK)=0
        XLOAD(1)=0
        DO 22 I=2,IJK-1
        XX=X(I)
        XLOAD(I)=(XX**CC)*((1 -XX)**CD)
22      CONTINUE
C
C GENERATE Y-COMPONENT GAP-AVERAGE VELOCITY
C
        XLOAD0=0
        CALL XINT(IJK,DX,XLOAD0,XLOAD,XILOAD)
        AMPV=(TAN2-TAN1)/XILOAD(1)
        DO 23 I=1,IJK
        VMY(I)=AMPV*XILOAD(I)+TAN1
        DVMY(I)=AMPV*XLOAD(I)
23      CONTINUE
        CALL DERIV(IJK,DX,DVMY,DDVMY)
C
C COMPUTE FUNCTIONS
C
        DO 2 I=1,IJK
        XX=X(I)
        RS(I)= 5*S-T(I)
        RI(I)=-(S*S/12 )+( 5*S*T(I))-( 5*T(I)**2)
        RJ(I)=-(S*S*T(I)/12 )+( 25*S*T(I)**2)-(T(I)**3/6 )
        VMX(I)= 5*S/RS(I)
        DVMX(I)=( 5*S*DT(I))/(RS(I)**2)
        DDVMX(I)=( 5*S*DDT(I)/RS(I)**2)+(S*DT(I)**2/RS(I)**3)
        DF(I)=VMY(I)/VMX(I)
        DDF(I)=(VMX(I)*DVMY(I)-VMY(I)*DVMX(I))/VMX(I)**2
2       CONTINUE
        FM0=0
        CALL XINT(IJK,DX,FM0,DF,FM)
C
C CALL SUBROUTINE TO COMPUTE EDGE VALUES
C
        DDTH=2 *DDT(1)/S
        RI0=RI(1)
        VMY1=VMY(1)
        DDVMX1=DDVMX(1)
        DDVMY1=DDVMY(1)
        FM1=FM(1)
        DF1=VMY(1)/VMX(1)
```

```
      F1=FM(1)
      CALL EDGE(ERR,NMAX,S,TAN1,TAN2,DDTH,RIO,VMY1,DDVMX1,DDVMY1
     1,FM1,F0,F1,DF0,DF1,A0,A1,B0,B1,CCR)
      FOLD(IJK)=F0
      FOLD(1)=F1
      DF(IJK)=DF0
      DF(1)=DF1
      F(IJK)=F0
      F(1)=F1
C
C GUESS F FOR ITERATION PURPOSE
C
      DELF0=-2 *FSUM(IJK,FOLD,T,RS,X,NMAX,COR)/S
      DO 130 I=2,IJK-1
      TERM1=RJ(I)*RS(I)*RI(I)
      TERM2=(-DVMY(I)-DF(I)*DVMX(I))/(1 *DF(I)**2)
      DELF=-2 *(TERM1*TERM2*FSUM(I,FOLD,T,RS,X,NMAX,COR))/S
      FOLD(I)=FM(I)*DELF
130   CONTINUE
      CALL DERIV(IJK,DX,FOLD,DF)
      DF(IJK)=DF0
      DF(1)=DF1
C
C CALL GUESSING SUBROUTINE FOR DBETA AND DDELT
C
      CALL GUESS(IJK,S,X,T,DVMX,DDVMX,DBETA,DDBETA,DDELT,DDDELT)
C
      WRITE(2,10)
10    FORMAT(5X,'GUESS INPUTS'//T10,'X',T25,'F',T40,'BETA',T55,'DELT'
     1,T70,'DXX',T85,'DYY'/)
      DO 11 J=1,IJK
      I=IJK+1-J
      DXX=DVMX(I)*DBETA(I)
      DYY=-DVMY(I)*DDELT(I)
      WRITE(2,12)X(I),FOLD(I),BETA(I),DELT(I),DXX,DYY
12    FORMAT(6(5X,F10 5))
11    CONTINUE
C
C ITERATION PROCESS FOR CAMBER LINE
C
      NNNN=0
      OLDERR=100
      OLDER=100
      DBETA(IJK)=-DVMX(IJK)
      DBETA(1)=-DVMX(1)
      DDELT(IJK)=DVMY(IJK)
      DDELT(1)=DVMY(1)
      DO 999 NNN=1,ITER
      NNNN=NNNN+1
C
      CALL HOMOB(NMAX,IJK,FOLD,T,RS,X,PHIHXB,PHIHYB,COR)
      CALL DERIV(IJK,DX,PHIHXB,DPHIHX)
      CALL DERIV(IJK,DX,PHIHYB,DPHIHY)
C
      DO 110 I=1,IJK
      DEN=1 *DF(I)**2
      A(I)=(DF(I)*(-DVMY(I)*DDELT(I))*DVMX(I)*DBETA(I))/DEN
110   CONTINUE
      CALL DERIV(IJK,DX,A,DA)
      CALL DERIV(IJK,DX,DA,DDA)
```

```
C
      DO 111 I=1,IJK
      DEN=1.+DF(I)**2
      B(I)=(DDVMY(I)-DDDELT(I)*A(I)*DDF(I)*2.*DA(I)*DF(I))/DEN
111   CONTINUE
      CALL DERIV(IJK,DX,B,DB)
C
      DO 112 I=2,IJK-1
      TERM=DT(I)*(RS(I)*RI(I)+RJ(I))/RS(I)**2
      DBETA(I)=RJ(I)*(DDA(I)-DB(I)*DF(I)-B(I)*DDF(I))/RS(I)
     1+TERM*(DA(I)-B(I)*DF(I))-DPHIHX(I)
      DDELT(I)=-RJ(I)*DB(I)/RS(I)-TERM*B(I)+DPHIHY(I)
112   CONTINUE
      CALL DERIV(IJK,DX,DBETA,DDBETA)
      CALL DERIV(IJK,DX,DDELT,DDDELT)
C
C COMPUTES CAMBER LINE
C
      DO 155 I=1,IJK
      OSC(I)=(-DVMY(I)+DDELT(I)-DF(I)*(DVMX(I)+DBETA(I)))/(1.+DF(I)**2)
      OPT(I)=OSC(I)
155   CONTINUE
      IF(NOPT EQ 1)CALL SMOOTH(IJK,IPOWER,X,AAAI,OSC,SMOSC)
C
      ERRMAX=0.
      SUMERR=0.
      XLOC=-1.
      DO 160 I=2,IJK-1
      TERM1=RJ(I)+RS(I)*RI(I)
      IF(NOPT EQ 1)OPT(I)=SMOSC(I)
      DELF=-2.*(TERM1*OPT(I)+FSUM(I,FOLD,T,RS,X,NMAX,COR))/S
      F(I)=FM(I)+DELF-DELF0
      ERROR=ABS(F(I)-FOLD(I))
      SUMERR=SUMERR+ERROR
      IF(ERROR GT ERRMAX)XLOC=X(I)
      IF(ERROR GT ERRMAX)ERRMAX=ERROR
160   CONTINUE
      CALL DERIV(IJK,DX,F,DF)
      DF(IJK)=DF0
      DF(1)=DF1
C
C WRITE CONVERGENCE
C
      IF(NOPT EQ 1)GO TO 165
      WRITE(1,162)NNNN,ERRMAX,XLOC
162   FORMAT(15X,'ITERATION #',I3,'------','ERRMAX =',F7.5
     1,'   AT X = ',F8.5)
      IF(ERRMAX GT (1.3*OLDERR))WRITE(1,163)AVGERR
163   FORMAT(//2X,'***** ITERATION SCHEME DIVERGES  !!!!! *****'/
     1       2X,'      AVERAGE ERROR = ',F9.5)
      IF(ERRMAX GT (1.3*OLDERR))GO TO 998
      IF(ERRMAX LE ERR)GO TO 998
      GO TO 169
C
165   ERRAVG=SUMERR/XIJK
      WRITE(1,166)NNNN,ERRAVG
166   FORMAT(15X,'ITERATION #',I3,'------','ERRAVG =',F7.5)
      IF(ERRAVG LE ERR)GO TO 998
C
C UPDATE VALUES
```

```
C
169     DO 170 I=2,IJK-1
        FOLD(I)=F(I)
170     CONTINUE
        OLDERR=ERRMAX
        OLDER=ERRAVG
C
999     CONTINUE
C
998     CONTINUE
C
C WRITE
C
        WRITE(1,60)
60      FORMAT(//T10,'X',T24,'LOAD(X)',T40,'T(X)',T55,'FM(X)',T70,'F(X)'/)
        DO 501 I=1,IJK
        J=IJK+1-I
        PLOT(1,I)=X(J)
        PLOT(2,I)=F(J)
        PLOT(3,I)=F(J)+T(J)
        PLOT(4,I)=F(J)-T(J)
        PLOT(5,I)=BLADET
        PLOT(6,I)=BLADEB
        TERM=AMPV*XLOAD(J)
        WRITE(1,61)X(J),TERM,T(J),FM(J),F(J)
61      FORMAT(5(5X,F10.5))
501     CONTINUE
        CALL QPICTR(PLOT,6,IJK,QY(2,3,4,5,6),QX(1),QLABEL(14)
        1,QYLAB('BLADE SHAPE'),QXLAB('AXIAL LOCATION X'))
C
        WRITE(2,70)
70      FORMAT(//5X,'OUTPUT'//T10,'X',T25,'F',T40,'BETA',T55,'DELT'
        1,T70,'DXX',T85,'DYY'/)
        DO 71 J=1,IJK
        I=IJK+1-J
        DXX=DVMX(I)+DBETA(I)
        DYY=-DVMY(I)+DDELT(I)
        IF(NOPT EQ 0)WRITE(4,75)DBETA(J),DDELT(J)
75      FORMAT(2(2X,F10.5))
        WRITE(2,72)X(I),F(I),BETA(I),DELT(I),DXX,DYY
72      FORMAT(5(5X,F10.5))
71      CONTINUE
C
C CALCULATE PRESSURE COEFFICIENT
C
        DO 800 I=1,IJK
        Y(I)=F(I)+T(I)
800     CONTINUE
        CALL VEL(IJK,NMAX,COR,X,Y,F,DF,DDF,RS,RI,RJ,PHIHXB,PHIHYB
        1,VMX,DVMX,DDVMX,VMY,DVMY,DDVMY,DBETA,DDBETA,DDELT,DDDELT
        1,VXT,VYT)
        DO 801 I=1,IJK
        Y(I)=F(I)+S-T(I)
801     CONTINUE
        CALL VEL(IJK,NMAX,COR,X,Y,F,DF,DDF,RS,RI,RJ,PHIHXB,PHIHYB
        1,VMX,DVMX,DDVMX,VMY,DVMY,DDVMY,DBETA,DDBETA,DDELT,DDDELT
        1,VXB,VYB)
        WRITE(1,799)
799     FORMAT(//T10,'X',T25,'Cp+',T40,'Cp-',T53,'(delta)P'/)
        DO 802 I=1,IJK
```

```
      VT=(VXT(I)**2+VYT(I)**2)** 5
      VB=(VXB(I)**2+VYB(I)**2)** 5
      CPT(I)=1 -(VT/VONSET)**2
      CPB(I)=1 -(VB/VONSET)**2
802   CONTINUE
C
810   DO 804 I=1,IJK
      J=IJK+1-I
      DIFF=ABS(CPT(J)-CPB(J))
      PLOT(2,I)=CPT(J)
      PLOT(3,I)=CPB(J)
      PLOT(4,I)=PRESST
      PLOT(5,I)=PRESSB
      WRITE(3,*)X(J),CPT(J),CPB(J),DIFF
      WRITE(1,805)X(J),CPT(J),CPB(J),DIFF
805   FORMAT(4(5X,F10 5))
804   CONTINUE
      CALL QPICTR(PLOT,6,IJK,QY(2,3,4,5),QX(1),QLABEL(14)
      1,QYLAB('PRESSURE COEFFICIENT Cp'),QXLAB('AXIAL LOCATION X'))
C
C WRITE BLADE SHAPE TO A FILE FOR USE IN THE NASA DIRECT METHOD
C
      DO 900 I=1,IJK
      WRITE(6,*)X(I)
900   CONTINUE
      DO 901 I=2,IJK
      J=IJK+1-I
      WRITE(6,*)X(J)
901   CONTINUE
      DO 902 I=1,IJK
      BSURF=F(I)-T(I)
      WRITE(6,*)BSURF
902   CONTINUE
      DO 903 I=2,IJK
      J=IJK+1-I
      TSURF=F(J)+T(J)
      WRITE(6,*)TSURF
903   CONTINUE
C
      STOP
      END
C
C ***************************************************************
C
C THIS FUNCTION COMPUTE FSUM(I)
C
      FUNCTION FSUM(I,FOLD,T,RS,X,MMAX,COR)
      REAL FOLD(101),T(101),RS(101),X(101)
      COMMON/COMM1/S,A0,A1,B0,B1,PI
      REAL LAMDM
      SUM=0
      DO 100 M=1,MMAX
      XM=M
      LAMDM=2 *PI*XM/S
      DELF0=(FOLD(I)-FOLD(IJK))*LAMDM
      DELF1=(FOLD(I)-FOLD(1))*LAMDM
      TTT=SIN(LAMDM*T(I))/LAMDM*RS(I)*COS(LAMDM*T(I))
      T0=TTT*EXP(-LAMDM*X(I))/LAMDM**3
      T1=TTT*EXP(-LAMDM*(1 -X(I)))/LAMDM**3
      TERM0=B0*COS(DELF0)+A0*SIN(DELF0)
```

```
      TERM1=-B1*COS(DELF1)+A1*SIN(DELF1)
      SUM=SUM+T0*TERM0+T1*TERM1
100   CONTINUE
      FSUM=COR*SUM
      RETURN
      END
C
C ************************************************************
C
      SUBROUTINE DERIV(IJK,DX,F,DF)
      REAL F(101),DF(101)
      DF(IJK)= 5*(-F(IJK-2)+4 *F(IJK-1)-3 *F(IJK))/DX
      DF(1)= 5*(3 *F(1)-4 *F(2)+F(3))/DX
      DO 1 I=2,IJK-1
      DF(I)= 5*(F(I-1)-F(I+1))/DX
1     CONTINUE
      RETURN
      END
C
C ************************************************************
C
      SUBROUTINE XINT(IJK,DX,XINTF0,F,XINTF)
      REAL F(101),XINTF(101)
      SUM=XINTF0
      XINTF(IJK)=SUM
      DO 1 I=2,IJK
      J=IJK+1-I
      SUM=SUM+ 5*(F(J)+F(J+1))*DX
      XINTF(J)=SUM
1     CONTINUE
      RETURN
      END
C
C ************************************************************
```

```
C
C SUBROUTINE EDGE   EDGE FOR
C THIS SUBROUTINE CALCULATE EXACT LE AND TE VALUES
C IN ORDER TO ACCELERATE CONVERGENCE IN F(X)
C
      SUBROUTINE EDGE(ERR,NMAX,S,TAN1,TAN2,DDTH,RIO,VMY1
      1,DDVMX1,DDVMY1,FM1,F0,F1,DF0,DF1,A0,A1,B0,B1,COR)
      REAL LAMDM
      PI=3 141592654
      F0=0
      DF0=TAN1
      IC=0
100   SUM01=0
      SUM11=0
      SUM02=0
      SUM22=0
      SUM03=0
      SUM33=0
      SUM00=0
      SUM=0
      DO 101 M=1,NMAX
      XM=M
      LAMDM=2 *PI*XM/S
      DELF=LAMDM*(F0-F1)
      CC=EXP(-LAMDM)/LAMDM**2
      SUMSIN=CC*SIN(DELF)
      SUMCOS=CC*COS(DELF)
      SUM01=SUM01+COR*SUMSIN
      SUM11=SUM11+COR*SUMCOS
      SUM02=SUM02+COR*SUMSIN/LAMDM
      SUM22=SUM22+COR*SUMCOS/LAMDM
      SUM03=SUM03+COR*SUMSIN*LAMDM
      SUM33=SUM33+COR*SUMCOS*LAMDM
      SUM00=SUM00+COR/LAMDM**3
      SUM=SUM+COR/LAMDM
101   CONTINUE
      Z1=RIO*DDTH-(1 -DF1**2)
      ZZZ=-(2 /RIO)*(SUM01**2-SUM11**2)
      Z2=(- 5*RIO+ZZZ)*DDTH
      Z22=( 5*RIO+ZZZ)*DDTH
      Z3=(4 /RIO)*SUM01*SUM11*DDTH
      SA1=((2 *DDTH*(SUM11+TAN1*SUM01))/RIO-DDVMX1)
      SA2=((-2 *DDTH*(SUM01-TAN1*SUM11))/RIO-DDVMY1)
      SA3=Z1-Z2
      SA4=2 *DF1+Z3
      SA22=Z1-Z22
      DEN=SA3*SA22+SA4*SA4
      A1=(SA1*SA22+SA2*SA4)/DEN
      B1=(SA2*SA3-SA1*SA4)/DEN
      A0=2 *(-A1*SUM11-B1*SUM01-1 )/RIO
      B0=2 *(A1*SUM01-B1*SUM11-TAN1)/RIO
      XNORM=-B0*SUM00-A1*SUM02+B1*SUM22
      FTE=FM1-B0*SUM22+A0*SUM02+B1*SUM00-XNORM
      DPHIHY=-SUM+B0*SUM33+B1-SUM03*A1
      DPHIHX=SUM*A0-SUM03*B1-SUM33*A1
      RATIO=(2 *DPHIHX-S*A0)/(DPHIHX**2- 25*S*S*A0)
      DFLE= 5*DPHIHY*RATIO
      XNUMTE=VMY1+ 75*B1*RIO+B0*SUM11+A0*SUM01
      DENTE=1 + 75*A1*RIO+B0*SUM01+A0*SUM11
      DFTE=XNUMTE/DENTE
```

```
ER1=ABS(FTE-F1)
ER2=ABS(DFTE-DF1)
ER3=ABS(DFLE-DF0)
F1=FTE
DF0=DFLE
DF1=DFTE
IC=IC+1
IF(ER1 GT ERR OR ER2 GT ERR OR ER3 GT ERR)GO TO 100
RETURN
END
```

```
C
C SUBROUTINE NAME   GUESS FOR
C THIS SUBROUTINE INITIALIZE BETA AND DELTA
C TO START THE ITERATION PROCESS OF F(X)
C
      SUBROUTINE GUESS(IJK,S,X,T,DVMX,DDVMX,DBETA,DDBETA,DDELT,DDDELT)
      REAL DBETA(101),DDBETA(101),DDELT(101),DDDELT(101),DVMX(101),DVMY(101)
     1,DDVMX(101),DDVMY(101),X(101),T(101)
C
      PI=3 141592654
      DO 1 I=1,IJK
      XX=X(I)
      TCOS=COS(2 *PI*XX)
      TSIN=SIN(2 *PI*XX)
      DBETA(I)=-DVMX(I)*TCOS
      DDBETA(I)=-DDVMX(I)*TCOS*2 *PI*DVMX(I)*TSIN
      DDELT(I)=0
      DDDELT(I)=0
1     CONTINUE
C
      RETURN
      END
```

```
C
C SUBROUTINE NAME   HOMOB FOR
C THIS SUBROUTINE CALCULATES THE HOMOGENEOUS
C TERMS PHIHXB AND PHIHYB
C
      SUBROUTINE HOMOB(NMAX,IJK,F,T,RS,X,PHIHXB,PHIHYB,COR)
      REAL F(101),X(101),T(101),RS(101),PHIHXB(101),PHIHYB(101)
      COMMON/COMM1/S,A0,A1,B0,B1,PI
      REAL LAMDM
      DO 200 I=1,IJK
      SUMX=0
      SUMY=0
      DO 100 M=1,NMAX
      XM=M
      LAMDM=2 *PI*XM/S
      DELF0=(F(I)-F(IJK))*LAMDM
      DELF1=(F(I)-F(1))*LAMDM
      XSIN=SIN(LAMDM*T(I))
      EXP0=EXP(-LAMDM*X(I))
      EXP1=EXP(-LAMDM*(1 -X(I)))
      SIN0=SIN(DELF0)
      SIN1=SIN(DELF1)
      COS0=COS(DELF0)
      COS1=COS(DELF1)
      T0=XSIN*EXP0/LAMDM**3
      T1=XSIN*EXP1/LAMDM**3
      SUMX=SUMX+T0*(B0*SIN0-A0*COS0)+T1*(B1*SIN1+A1*COS1)
      SUMY=SUMY+T0*(-B0*COS0-A0*SIN0)+T1*(-B1*COS1+A1*SIN1)
100   CONTINUE
      PHIHXB(I)=COR*SUMX/RS(I)
      PHIHYB(I)=COR*SUMY/RS(I)
200   CONTINUE
      RETURN
      END
```

```
C
C SUBROUTINE NAME   VEL FOR
C THIS SUBROUTINE COMPUTES THE VELOCITY FIELD
C IN THE CASCADE REGION (REGION 2)
C
      SUBROUTINE VEL(IJK,NMAX,COR,X,Y,F,DF,DDF,RS,RI,RJ,PHIHXB,PHIHYB
     1,VMX,DVMX,DDVMX,VMY,DVMY,DDVMY,DBETA,DDBETA,DDELT,DDDELT,VX,VY)
      REAL X(101),Y(101),F(101),DF(101),DDF(101),RS(101),RI(101),RJ(101)
     1,PHIHXB(101),PHIHYB(101),VMX(101),DVMX(101),DDVMX(101),VMY(101)
     1,DVMY(101),DDVMY(101),DBETA(101),DDBETA(101),DDELT(101),DDDELT(101)
     1,VX(101),VY(101),TVX(101),TVY(101),AX(101),BX(101)
     1,PHIHX(101),PHIHY(101)
      COMMON/COMM1/S,A0,A1,B0,B1,PI
C
      CALL HOMO(NMAX,IJK,F,X,Y,PHIHX,PHIHY,COR)
C
      AX(IJK)=A0
      AX(1)=A1
      BX(IJK)=B0
      BX(1)=B1
      DO 200 I=2,IJK-1
      DEN=(1 +DF(I)**2)**3
      X1=2 *DF(I)*(1 +DF(I)**2)
      X2=1 -DF(I)**4
      X3=DDF(I)*(1 -3 *DF(I)**2)
      X4=DF(I)*DDF(I)*(3 -DF(I)**2)
      DXX=DVMX(I)+DBETA(I)
      DDXX=DDVMX(I)+DDBETA(I)
      DYY=-DVMY(I)+DDELT(I)
      DDYY=-DDVMY(I)+DDDELT(I)
      AX(I)=(X1*DDYY+X2*DDXX+X3*DYY-X4*DXX)/DEN
      BX(I)=(-X2*DDYY+X1*DDXX+X4*DYY+X3*DXX)/DEN
      TVX(I)=(DYY-DF(I)*DXX)/(1 +DF(I)**2)
      TVY(I)=(DXX+DF(I)*DYY)/(1 +DF(I)**2)
200   CONTINUE
C
      DO 100 I=1,IJK
      ALPHA=Y(I)-F(I)
      RSY= S*S-ALPHA
      RIY=-(S*S/12 )+( S*S*ALPHA)-( S*ALPHA**2)
      CCC=RJ(I)/RS(I)
      TERMX=RSY*TVX(I)+(CCC+RIY)*AX(I)
      TERMY=RSY*TVY(I)+(CCC+RIY)*BX(I)
      VX(I)=VMX(I)+PHIHX(I)-PHIHXB(I)+TERMX
      VY(I)=VMY(I)+PHIHY(I)-PHIHYB(I)+TERMY
100   CONTINUE
C
      RETURN
      END
```

```
C
C SUBROUTINE NAME   HOMO FOR
C THIS SUBROUTINE CALCULATES THE HOMOGENEOUS
C TERMS PHIHX AND PHIHY
C
      SUBROUTINE HOMO(NMAX,IJK,F,X,Y,PHIHX,PHIHY,COR)
      REAL F(101),X(101),Y(101),PHIHX(101),PHIHY(101)
      COMMON/COMM1/S,A0,A1,B0,B1,PI
      REAL LAMDM
      DO 200 I=1,IJK
      SUMX=0
      SUMY=0
      DO 100 M=1,NMAX
      XM=M
      LAMDM=2 *PI*XM/S
      DELF0=(Y(I)-F(IJK))*LAMDM
      DELF1=(Y(I)-F(1))*LAMDM
      EXP0=EXP(-LAMDM*X(I))
      EXP1=EXP(-LAMDM*(1 -X(I)))
      SIN0=SIN(DELF0)
      SIN1=SIN(DELF1)
      COS0=COS(DELF0)
      COS1=COS(DELF1)
      T0=EXP0/LAMDM**2
      T1=EXP1/LAMDM**2
      SUMX=SUMX+T0*(-B0*SIN0+A0*COS0)+T1*(B1*SIN1+A1*COS1)
      SUMY=SUMY+T0*(B0*COS0+A0*SIN0)+T1*(B1*COS1-A1*SIN1)
100   CONTINUE
      PHIHX(I)=COR*SUMX
      PHIHY(I)=COR*SUMY
200   CONTINUE
      RETURN
      END
```

```
C
C SUBROUTINE NAME   SMOOTH FOR
C THIS SUBROUTINE SMOOTH A GIVEN FUNCTION F(X)
C USING A MODIFIED LEAST-SQUARE SCHEME
C
      SUBROUTINE SMOOTH(IJK,IORDER,X,AI,F,SMF)
      REAL X(101),F(101),SMF(101),AI(10,10),B(10),SMCF(10)
C
      DO 400 J=1,IORDER-1
      SUM=0
      DO 500 I=1,IJK
      SUM=SUM+(F(I)-F(IJK))*X(I)**J
500   CONTINUE
      B(J)=SUM
400   CONTINUE
      B(IORDER)=F(1)-F(IJK)
C
      DO 100 I=1,IORDER
      SUM=0
      DO 200 J=1,IORDER
      SUM=SUM+AI(I,J)*B(J)
200   CONTINUE
      SMCF(I)=SUM
100   CONTINUE
C
      DO 600 I=1,IJK
      SUM=F(IJK)
      DO 700 J=1,IORDER
      SUM=SUM+SMCF(J)*(X(I)**J)
700   CONTINUE
      SMF(I)=SUM
600   CONTINUE
C
      RETURN
      END
```

```
C
C SUBROUTINE NAME   SMAI FOR
C SUBROUTINE USED TO GENERATE MATRIX [ AI ]
C USED IN THE LEAST-SQUARE FITTING FILTER
C
      SUBROUTINE SMAI(IJK,IORDER,X,AI)
      REAL AI(10,10),B(10),WKAREA(20),X(101),XM(19)
C
      DO 100 I=2,2*IORDER
      SUM=0
      DO 200 J=1,IJK
      SUM=SUM+X(J)**I
200   CONTINUE
      XM(I-1)=SUM
100   CONTINUE
C
      K=0
      DO 10 I=1,IORDER
      DO 20 J=1,IORDER
      AI(I,J)=XM(K+J)
20    CONTINUE
      K=K+1
10    CONTINUE
      DO 300 J=1,IORDER
      AI(IORDER,J)=1
300   CONTINUE
C
      D1=-1
      CALL LINV3F(AI,B,1,IORDER,10,D1,D2,WKAREA,IER)
C
      RETURN
      END
```

```
C
C SUBROUTINE NAME   FILTER FOR
C THIS SUBROUTINE FILTER THE PRESSURE
C COEFFICIENTS
C
      SUBROUTINE FILTER3(IJK,X,F,FFILT)
      REAL X(101),F(101),FT(101),XT(101),FB(101),XB(101),FFILT(101)
      XIJK=IJK-1
      PI=3 141592654
      DX=1 /XIJK
      IJ=(IJK+1)/2
C
      NB=0
      NT=0
      I=IJ-1
100   I=I+1
      IF(I GT (IJK-1))GO TO 999
      IF(F(I+1) GT F(I))GO TO 100
      IF(I EQ IJ)GO TO 20
      NT=NT+1
      FT(NT)=F(I)
      XT(NT)=X(I)
200   I=I+1
      IF(I GT (IJK-1))GO TO 999
20    IF(F(I+1) LE F(I))GO TO 200
      NB=NB+1
      FB(NB)=F(I)
      XB(NB)=X(I)
      GO TO 100
999   CONTINUE
      FT(NT+1)=F(IJK)
      FB(NB+1)=F(IJK)
      XT(NT+1)=X(IJK)
      XB(NB+1)=X(IJK)
C
      NP=NT+1
      CALL FIT(IJK,DX,NP,X,XT,F,FT)
      NP=NB+1
      CALL FIT(IJK,DX,NP,X,XB,F,FB)
      DO 700 I=1,IJK
      FFILT(I)= 5*(FB(I)+FT(I))
700   CONTINUE
C
      RETURN
      END
C
C ************************************************************
C
      SUBROUTINE FIT(IJK,DX,NP,X,XTB,F,FTB)
      REAL X(101),XTB(101),F(101),FTB(101),FFIT(101)
      1,XFILT(101),FILT1(101),FILT2(101),AAAI(10,10)
C
      XNST=XTB(1)/DX
      NST=IJK-IIFIX(XNST)
      NST1=NST
      DO 10 I=2,NP
      XM=(FTB(I)-FTB(I-1))/(XTB(I)-XTB(I-1))
      XB=FTB(I)-XM*XTB(I)
      XJ=XTB(I)/DX
      JJ=IJK-IIFIX(XJ)
```

```
      DO 20 N=NST,JJ
      FFIT(N)=XM*X(N)+XB
20    CONTINUE
      NST=JJ
10    CONTINUE
C
      DO 30 I=1,NST1
      FTB(I)=F(I)
30    CONTINUE
      DO 31 I=NST1+1,IJK
      FTB(I)=FFIT(I)
31    CONTINUE
C
      N=0
      DO 100 I=NST1,IJK-2
      N=N+1
      XFILT(N)=X(I)
      FILT1(N)=FTB(I)
100   CONTINUE
      CALL SMAI(N,4,XFILT,AAAI)
      CALL SMOOTH(N,4,XFILT,AAAI,FILT1,FILT2)
      N=0
      DO 101 I=NST1,IJK-2
      N=N+1
      FTB(I)=FILT2(N)
101   CONTINUE
C
      DO 200 I=NST1-3,IJK-3
      FTB(I)= 3333*(FTB(I+1)+FTB(I)+FTB(I-1))
200   CONTINUE
      DO 201 I=NST1-3,IJK-5
      FTB(I)= 3333*(FTB(I+2)+FTB(I)+FTB(I-2))
201   CONTINUE
C
      RETURN
      END
C
C ***************************************************
```

END

FILMED

4-85

DTIC